

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re application of: Kimmo MYLLY

Group No. :

Serial No.:

Filed: Herewith

Examiner:

For: A METHOD FOR STORING DATA IN A MEMORY, A SYSTEM, AN ELECTRONIC DEVICE AND A MEMORY CARD

Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

TRANSMITTAL OF CERTIFIED COPY

Attached please find the certified copy of the foreign application from which priority is claimed for this case, along with the certified translation of Tuulikki Tulivirta, Certified Translator:

Country :Finland
Application Number :20035041
Filing Date :March 31, 2003

WARNING: "When a document that is required by statute to be certified must be filed, a copy, including a photocopy or facsimile transmission of the certification is not acceptable." 37 CFR 1.4(f) (emphasis added.)

Francis Maxine

SIGNATURE OF ATTORNEY

Reg. No.: 31,391

Francis J. Maguire

Type or print name of attorney

Tel. No.: (203) 261-1234

Ware, Fressola, Van Der Sluys &
Adolphson LLP
755 Main Street, P.O. Box 224
P.O. Address

Customer No.: 004955

Monroe, Connecticut 06468

NOTE: The claim to priority need be in no special form and may be made by the attorney or agent if the foreign application is referred to in the oath or declaration as required by § 1.63.

CERTIFICATE OF MAILING (37 CFR 1.10)

I hereby certify that this correspondence is, on the date shown below, being deposited with the United States Postal Service as Express Mail No.: EV 435647459US in an envelope addressed to the: Mail Stop Patent Application, Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450.

Date: March 30, 2004

Annemarie Maher

(Type or print name of person certifying)

Answer Male

(Signature of person mailing paper)

(Transmittal of Certified Copy [5-4])

PATENTTI- JA REKISTERIHALLITUS
NATIONAL BOARD OF PATENTS AND REGISTRATION

Helsinki 4.3.2004

ETUOIKEUSTODISTUS
PRIORITY DOCUMENT



Hakija
Applicant

Nokia Corporation
Helsinki

Patenttihakemus nro
Patent application no

20035041

Tekemispäivä
Filing date

31.03.2003

Kansainvälinen luokka
International class

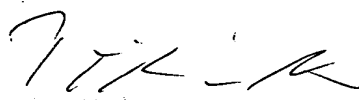
G06F

Keksinnön nimitys
Title of invention

**"Menetelmä tiedon tallentamiseksi muistiin, järjestelmä,
elektroniikkalaite ja muistikortti"**

Täten todistetaan, että oheiset asiakirjat ovat tarkkoja jäljennöksiä Patentti- ja rekisterihallitukselle alkuaan annetuista selityksestä, patenttivaatimuksista, tiivistelmästä ja piirustuksista.

This is to certify that the annexed documents are true copies of the description, claims, abstract and drawings originally filed with the Finnish Patent Office.


Pirjo Kaila
Tutkimussihteeri

Maksu 50 €
Fee 50 EUR

Maksu perustuu kauppa- ja teollisuusministeriön antamaan asetukseen 1027/2001 Patentti- ja rekisterihallituksen maksullisista suoritteista muutoksineen.

The fee is based on the Decree with amendments of the Ministry of Trade and Industry No. 1027/2001 concerning the chargeable services of the National Board of Patents and Registration of Finland.

Osoite:	Arkadiankatu 6 A	Puhelin:	09 6939 500	Telefax:	09 6939 5328
	P.O.Box 1160	Telephone:	+ 358 9 6939 500	Telefax:	+ 358 9 6939 5328
	FIN-00101 Helsinki, FINLAND				

Menetelmä tiedon tallentamiseksi muistiin, järjestelmä, elektroniikka-laitte ja muistikortti

- 5 Nyt esillä oleva keksintö kohdistuu menetelmään muistin käsittelytoiminnon suorittamiseksi lohkomuistissa, jossa on muistisoluja, joihin tietoa voidaan tallentaa, ja väyläliitäntä, jossa on ainakin tilalinja, joka on asetettavissa ainakin valmiina-tilaan ja toiminto kesken -tilaan, ja jossa menetelmässä suoritetaan käsittelytoiminnon alussa mainitun tilalinjan tilan asettaminen mainittuun toiminto kesken -tilaan, ja suoritetaan käsittelyvirheiden havainnointi, ja käsittelytoiminnon päättymisen ilmaistaan asettamalla mainittu tilalinja mainittuun valmiina-tilaan. Keksintö kohdistuu lisäksi järjestelmään, joka käsittää elektroniikkalaitteen, jossa on lohkomuistia, jossa on muistisoluja, joihin tietoa on
- 10 järjestetty tallennettavaksi, ja väyläliitäntä, jossa on ainakin tilalinja, joka on järjestetty asetettavaksi ainakin valmiina-tilaan ja toiminto kesken -tilaan, ja joka lohkomuisti käsittää välineet mainitun tilalinjan tilan asettamiseksi mainittuun toiminto kesken -tilaan lohkomuistin käsittelytoiminnon alussa, vertailijan käsittelyvirheiden havaitsemiseksi, ja käsittelytoiminnon päättymisen on järjestetty ilmaistavaksi asettamalla mainittu tilalinja mainittuun valmiina-tilaan. Keksintö kohdistuu lisäksi elektroniikkalaitteeseen, joka käsittää lohkomuistia, jossa on muistisoluja, joihin tietoa on järjestetty tallennettavaksi, ja väyläliitäntä, jossa on ainakin tilalinja, joka on järjestetty asetettavaksi ainakin valmiina-tilaan ja toiminto kesken -tilaan, ja joka lohkomuisti käsittää välineet mainitun tilalinjan tilan asettamiseksi mainittuun toiminto kesken -tilaan lohkomuistin käsittelytoiminnon alussa, vertailijan käsittelyvirheiden havaitsemiseksi, ja käsittelytoiminnon päättymisen on järjestetty ilmaistavaksi asettamalla mainittu tilalinja mainittuun valmiina-tilaan. Keksintö kohdistuu vielä lohkomuistiin, jossa on muistisoluja, joihin tietoa on järjestetty tallennettavaksi, ja väyläliitäntä, jossa on ainakin tilalinja, joka on järjestetty asetettavaksi ainakin valmiina-tilaan ja toiminto kesken -tilaan, ja joka lohkomuisti käsittää välineet mainitun tilalinjan tilan asettamiseksi mainittuun toiminto kesken -tilaan lohkomuistin käsittelytoiminnon alussa, vertailijan käsittelyvirheiden havaitsemiseksi, ja käsittelytoiminnon päättymisen on järjestetty ilmaistavaksi asettamalla mainittu tilalinja mainittuun valmiina-tilaan.
- 20
- 25
- 30
- 35

- Tunnetaan muisteja, joissa tiedon tallentaminen muistiin ja tallennetun tiedon lukeminen suoritetaan tavujen/sanojen sijasta suuremmissa yksiköissä. Esimerkiksi NAND-muistit ja NOR-muistit koostuvat muistisolumatriiseista. Kukin tällaisen matriisin rivi muodostaa yhden sivun (page). Näistä riveistä voidaan koota suurempia kokonaisuuksia (lohkoja, block), jolloin osa muistioperaatioista on sivukohtaisia ja osa kohdistuu yhteen tai useampaan lohkokoon. Sivun koko on esimerkiksi eräässä 64 megatavun NAND-muistissa 528 bittiä, joista 16 bittiä on normaalisti vain varalla, eli yhdellä sivulla käytettävissä olevien bittien määrä on 512. Yksi lohko koostuu tässä esimerkkimuistissa 16:ta sivusta, mutta myös muita lohkokokoja voidaan käyttää. Tällaisista muisteista käytetään tässä selityksessä myös nimitystä lohkomuisti.
- NAND-muisteissa ei ole osoiteväylää, vaan osoitteiden välitys muistiin suoritetaan dataväylän kautta. Dataväylän leveys on tavallisesti 8 bittiä. Tyypillisesti NAND-muisteissa tietojen tallentaminen muistiin ja tietojen lukeminen muistista suoritetaan sivu kerrallaan, mutta esim. muistin tyhjentäminen (muistisolujen asettaminen tiettyyn arvoon, joko 0-tilaan tai 1-tilaan) suoritetaan lohkoittain. Yksittäisen tavun tietojen muuttaminen edellyttää koko sen sivun tietojen uudelleenkirjoittamista, jossa muutettava tavu sijaitsee. Tällaisen muistin käyttö on jonkin verran hankalampaa kuin tavanomaisen satunnaissaantimuistin (RAM, Random Access Memory) käyttö. Kun tällaiseen muistiin kirjoitetaan tietoa, toimitaan tyypillisesti siten, että laitteen suoritin lähettää tallennettavan tiedon sivu kerrallaan muistiohjaimelle, joka suorittaa tiedon edelleenlähetyksen muistiin. Muistille ilmoitetaan ensin tallennuspaikan osoite, tavallisesti sivun alkuosoite. Tämä tapahtuu siten, että I/O-väylään asetetaan tätä tarkoittava komento, minkä jälkeen komentolinjaan asetetaan pulssi, jolloin muisti käy lukemassa I/O-väylässä olevan komennon. Tällä tavoin annetaan komennon lisäksi myös tallennuspaikan osoite. Seuraavassa vaiheessa aloitetaan tallennettavan sivun tietojen lähettäminen muistiohjaimelta muistiin tyypillisesti tavuittain. Muistissa on välimuistia, johon muistiin tallennettava sivu kirjoitetaan.
- Tietojen tallennuksen jälkeen suoritetaan tallennuksen oikeellisuuden tarkistus. Eräissä tunnetun tekniikan mukaisissa muisteissa tämä suo-

ritetaan automaattisesti jokaisen tallennetun sivun jälkeen vertaamalla tallentunutta tietoa tallennettavana olleeseen tietoon. Vertailun tila tallennetaan muistin sisäiseen tilarekisteriin. Tämän tilarekisterin tila voidaan tallennuksen jälkeen tarkistaa lähettämällä muistille tilarekisterin lukukomento, johon muisti palauttaa vasteena tilarekisterin sisällön. Tällöin muistiohjain voi tämän tilatiedon perusteella päätellä sen, tallentuivatko tiedot oikein. Tallennuksen tilatiedon lähettämiseksi muistiohjain välittää esim. keskeytyspyynnön suorittimelle, joka tämän perusteella käy lukemassa muistiohjaimen lähettämän tilatiedon. Tällaisessa järjestelyssä on mm. se epäkohta, että suorittimelle tulee keskeytyksiä jokaisen sivun käsittelyn jälkeen, vaikka tiedot olisivatkin tallentuneet oikein muistiin. Tallennettava tietomäärä voi olla hyvinkin suuri, jolloin keskeytyksiä on runsaasti, mikä hidastaa suorittimen toimintaa.

Edellä esitetyn kaltaisia muistirakenteita on useita eri tekniikkoihin perustuvia, kuten NAND, XOR jne. Yhteistä näille muisteille on kuitenkin se, että niissä tieto pysyy käyttöjännitteiden katkeamisen jälkeen, eli ne ovat ns. haihtumattomia muisteja. Lisäksi tietojen tallentaminen on hitaampaa tällaisiin haihtumattomiin muisteihin kuin dynaamisiin ja staattisiin luku/kirjoitusmuisteihin.

Haihtumattomia muisteja voidaan järjestää muistikorteiksi, jotka voidaan irrotettavasti kiinnittää elektroniikkalaitteeseen. Tällaisia muistikortteja voidaan tällöin siirtää laitteesta toiseen, mikä mahdollistaa mm. tietojen siirron laitteiden välillä. Esimerkiksi digitaalikamerassa kuvia voidaan tallentaa muistikorttiin. Tallentamisen jälkeen muistikortti voidaan irrottaa digitaalikamerasta ja asettaa esim. tietokoneeseen tai kannettavaan viestintälaitteeseen. Tällöin kuvat voidaan lukea muistikortista tämän toisen laitteen muistiin. Muita muistikorttien sovelluksia on mm. ohjelmatiedostojen, videoiden, musiikin, tekstitiedostojen yms. tallennus.

Vaikka edellä on esitetty, että muistin käsittelyä varten on järjestetty muistiohjain, jonka kautta tietojen välitys suorittimen ja muistin välillä suoritetaan, on selvää, että suoritin voi suoraankin ohjata muistia. Täl-

löin kuitenkin muistinkäsittelyn aiheuttama kuormitus suorittimessa on vielä suurempi kuin muistiohjainta käytettäessä.

- 5 Nyt esillä olevan keksinnön eräänä tarkoituksena on aikaansaada parannettu menetelmä ja järjestelmä tietojen tallentamiseksi muistiin, sekä menetelmää soveltava muisti ja laite. Keksintö perustuu siihen ajatukseen, että suorittimelle välitetään tieto tallennuksesta vain siinä tapauksessa, että tallennuksessa on tullut virhe. Tämä tiedon välittäminen voidaan järjestää edullisesti keskeytyksellä, joka voidaan välittää
- 10 jotakin muistin liitäntärajapinnassa olemassa olevaa linjaa tai tähän tarkoitukseen muodostettua linjaa pitkin. Täsmällisemmin ilmaistuna nyt esillä olevan keksinnön mukaiselle menetelmälle on pääasiassa tunnusomaista se, että mikäli havaitaan käsittelyvirhe, mainitun tilalinjan tila muutetaan takaisin mainittuun toiminto kesken –tilaan. Keksinnön mukaiselle järjestelmälle on pääasiassa tunnusomaista se, että käsittelyvirheen havaitsemisen jälkeen mainitun tilalinjan tila on järjestetty muutettavaksi takaisin mainittuun toiminto kesken –tilaan. Keksinnön mukaiselle elektroniikkalaitteelle on pääasiassa tunnusomaista se, että käsittelyvirheen havaitsemisen jälkeen mainitun tilalinjan tila on järjestetty muutettavaksi takaisin mainittuun toiminto kesken –tilaan. Keksinnön mukaiselle lohkomuistille on pääasiassa tunnusomaista se, että käsittelyvirheen havaitsemisen jälkeen mainitun tilalinjan tila on järjestetty muutettavaksi takaisin mainittuun toiminto kesken –tilaan.
- 20
- 25 Keksinnön erään edullisen suoritusmuodon mukaisessa järjestelmässä ja elektroniikkalaitteessa muistiohjain suorittaa muistiosoitteiden kasvattamisen, mikäli tietoja tallennetaan peräkkäin useammalle kuin yhdelle sivulle.
- 30 Nyt esillä olevalla keksinnöllä saavutetaan merkittäviä etuja tunnetun tekniikan mukaisiin ratkaisuihin verrattuna. Keksinnön mukaisella menetelmällä voidaan vähentää suorittimen kuormitusta, koska suorittimelle ei aiheuteta keskeytystä sellaisissa tilanteissa, joissa tallennus on suoritettu virheettömästi. Lisäksi muistiohjaimen ja muistin välinen
- 35 komentojen välitystarve on pienempi kuin tunnetun tekniikan mukaisia muistirakenteita käytettäessä. Keksinnön erään edullisen suoritusmuodon mukaisella menetelmällä saavutetaan vielä se etu, että suorittimen

ei tarvitse huolehtia muistiosoitteiden kasvattamisesta, jolloin suorittimen suoritusaikaa vapautuu muiden toimenpiteiden suorittamiseen.

Seuraavassa keksintöä selostetaan tarkemmin viitaten samalla oheisiin
5 piirustuksiin, joissa

- kuva 1 esittää keksinnön erään edullisen suoritusmuodon mukaista
 elektroniikkalaitetta pelkistettynä kaaviona,
- 10 kuva 2a esittää keksinnön erään edullisen suoritusmuodon mukai-
 sessa menetelmässä käytettävää suorittimen, muistiohjai-
 men ja muistin välistä signalointia muistin tyhjennyksen yh-
 teydessä,
- 15 kuva 2b esittää keksinnön erään edullisen suoritusmuodon mukai-
 sessa menetelmässä käytettävää suorittimen, muistiohjai-
 men ja muistin välistä signalointia muistiin tallennuksen yh-
 teydessä, ja
- 20 kuva 3 esittää keksinnön erään edullisen suoritusmuodon mukaista
 lohkomuistia pelkistettynä kaaviona.

Seuraavassa keksinnön erään edullisen suoritusmuodon mukaisen
25 menetelmän kuvauksessa käytetään esimerkkinä kuvan 1 mukaista
elektroniikkalaitetta 1, jossa on mm. lohkomuistia 1, kuten NAND-muis-
tia tai NOR-muistia. Lohkomuistin 2 ohjaamista varten on elektroniik-
kalaitteeseen 1 edullisesti järjestetty muistiohjain 3. On kuitenkin sel-
vää, että muistiohjainta 3 ei tarvita kaikissa sovelluksissa, vaan esime-
30 riksi suoritinta 4 voidaan käyttää muistinhallintatoiminnoissa. Tässä
suoritusmuodossa muistiohjain 3 ja suoritin 4 on järjestetty samaan so-
velluskohtaisesti ohjelmoitavaan integroituun piiriin 6 (ASIC,
Application Specific Integrated Circuit), mutta nämä voivat olla myös
erillisiä piirejä.

35 Lohkomuistin 2 ja muistiohjaimen 3 välinen liitântäväylä 5 käsittää
edullisesti ainakin yhdestä datalinjasta koostuvan dataväylän 5a (data)
sekä ohjauslinjoja 5b—5h. Näiden ohjauslinjojen avulla voidaan hallita

muistitoimintojen suorittamista. Tämän edullisen suoritusmuodon mukaisessa elektroniikkalaitteessa ohjauslinjat käsittävät komennon liipaisulinjan 5b (CLE, Command Latch Enable), osoitteen liipaisulinjan 5c (ALE, Address Latch Enable), piirivalintalinjan 5d (CE, Chip Enable), lukulinjan 5e (RE, Read Enable), kirjoituslinjan 5f (WE, Write Enable), kirjoituksen estolinjan 5g (WP, Write Protect), ja tilalinjan 5h (R/B, Ready/Busy). Kuvassa 1 joidenkin ohjauslinjojen yhteydessä oleva #-merkintä tarkoittaa sitä, että kyseinen linja on ns. alhaalla aktiivinen, eli loogisessa 0-tilassa aktiivinen. On kuitenkin selvää, että keksintöä ei ole rajoitettu ainoastaan kuvan 1 mukaisiin vaihtoehtoihin, vaan myös ylhäällä (loogisessa 1-tilassa) aktiivisia linjoja voidaan käyttää.

Suorittimen 4 ja muistiohjaimen 3 välillä on myös signaalien välitystä varten väyläliityntä 7. Tämän väyläliitynnän kautta välitetään mm. osoitteita, dataa ja ohjaustietoa, kuten keskeytyksiä suorittimen 4 ja muistiohjaimen 3 välillä.

Keksinnön erään edullisen suoritusmuodon mukaisessa menetelmässä lohkomuistin 2 hallinta suoritetaan edullisesti seuraavasti. Kuvataan ensin lohkomuistin 2 tyhjennystä lohkoittain. Tätä on esitetty kuvassa 2a signalointikaaviona. Suoritin 4 lähettää muistiohjaimelle 3 sen toimintatilaan vaikuttavia toimintaparametreja (nuoli 201 kuvan 2a kaaviossa). Tämän jälkeen suoritin 4 lähettää muistiohjaimelle 3 muistilohkon tyhjentämiskomennon sekä ilmoittaa tyhjennettävän lohkon alkusoitteen, järjestysnumeron tai muun vastaavan tiedon, jolla tyhjennettävä lohko lohkomuistissa 2 käy selville (nuoli 202). Lisäksi muistiohjaimelle 3 ilmoitetaan edullisesti lohkojen lukumäärä tai tyhjennettävän muistialueen pituus, jos tyhjennettäviä lohkoja on useampia kuin yksi. Tämän jälkeen muistiohjain 3 asettaa tarvittaessa piirivalintalinjan 5d tilaan, jolla lohkomuistin väyläliitântä on aktiivisena. Tämä tila on esimerkiksi looginen 0-tila. Kun lohkomuisti 2 on aktivoitu, tutkii muistiohjain 3 ensin sen, onko lohkomuisti 2 valmiina esim. komentojen vastaanottoon vai onko lohkomuistissa 2 jokin toiminto kesken. Jos lohkomuisti 2 on suorittamassa esim. tietojen tallennusta tai muistin tyhjennystä, lohkomuisti 2 ei tavallisesti voi vastaanottaa uusia komentoja. Tämä voidaan selvittää tutkimalla tilalinjan 5h tila. Tila voi olla joko valmiina-tila tai toiminto kesken -tila. Oletetaan tässä, että val-

miina-tilassa tilalinja 5h on loogisessa 1-tilassa, ja vastaavasti toiminto kesken –tilassa tilalinja on loogisessa 0-tilassa.

5 Tilalinjan 5h ilmaistessa, että lohkomuisti 2 on valmis vastaanottamaan komentoja, asettaa muistiohjain 3 liitântäväylän 5 dataväylälle 5a lohkon tyhjennyskomennon (nuoli 203). Tämän jälkeen komennon liipaisulinjalle 5b lähetetään edullisesti pulssi, jolloin lohkomuistin 2 sisäinen ohjain 16 lukee komennon dataväylältä 5a pulssin reunalla. Tämä reuna voi sovelluksesta riippuen olla joko nouseva reuna tai laskeva reuna. Seuraavaksi muistiohjain 3 asettaa dataväylälle 5a tyhjennettävän lohkon osoitetiedon tai osan siitä ja sen jälkeen kun osoitetieto on asettunut dataväylälle 5a, muistiohjain 3 muodostaa pulssin osoitteen liipaisulinjaan 5c. Tällöin lohkomuistin ohjain 16 lukee osoitetiedon dataväylältä 5a. Tässä esimerkissä käytettävällä tyypillisellä 8-bittisellä 10 väylällä ei voida osoittaa koko lohkomuistin 2 muistiavaruutta, jolloin osoite on annettava useammassa osassa. Tällöin muistiohjain 3 toistaa edellä kuvattua osoitteenvälitystä tarvittavan määrän, tyypillisesti kaksi tai kolme kertaa, jolloin lohkomuisti 2 käyttää kaikkia välitettyjä osoitetietoja tyhjennettävän lohkon yksilöimiseksi. Kun koko osoite on välitetty lohkomuistiin 2, vaihtaa lohkomuisti 2 (edullisesti ohjain 16) tilalinjan 5h tilan, eli asettaa sen toiminto kesken –tilaan, esim. loogiseen 0-tilaan. Tämän jälkeen lohkomuistissa 2 aloitetaan muistialueen tyhjentämisen, eli kaikkien tämän muistialueen bittien asettamisen tiettyyn 20 tilaan, edullisesti loogiseen 1-tilaan. Kun tilalinjan 5h tila on toiminto kesken –tilassa, on lohkomuisti 2 suorittamassa asetettua toimintoa, eli 25 tässä tapauksessa muistin tyhjennystä.

Muistialueen tyhjentämisen yhteydessä lohkomuistissa 2 tarkistetaan, onnistuiko kaikkien bittien tyhjentäminen vai ei. Mikäli jokin bitti ei 30 asettunut haluttuun tilaan, tallennetaan tilatieto lohkomuistissa 2. Tyhjennys- ja vertailutoiminnon suorittamisen jälkeen tilalinjan 5h tila palautetaan valmiina-tilaan, esim. loogiseen 1-tilaan (nuoli 204). Tämä tilalinjan 5h tilan vaihtuminen aiheuttaa muistiohjaimella 3 keskeytyksen, jolloin muistiohjain 3 siirtyy suorittamaan vastaavaa keskeytyspalveluohjelmaa. Jos tyhjennyksessä tapahtui virhe, välitetään tästä virhetieto muistiohjaimelle 3 edullisesti seuraavasti. Lohkomuistissa 2 35 vaihdetaan tilalinjan 5h tila takaisin toiminto kesken –tilaan. Nyt muisti-

ohjain 3 lukee tilalinjan 5h tilan (lohko 205), ja jos se on mainitussa toiminto kesken –tilassa, voi muistiohjain 3 päätellä virheen tapahtuneen. Tästä virheestä voidaan tarvittaessa ilmoittaa suorittimelle 4 esimerkiksi keskeytyksellä. Jos tilalinjan 5h tila sen sijaan on valmiina-

5 tilassa, voi muistiohjain 3 päätellä toiminnon onnistuneen ja seuraava komento voidaan välittää lohkomuistiin 2 (nuoli 206). Tällä järjestelyllä muistiohjaimen 3 ei tarvitse käydä erikseen lukemassa tilatietoa lohkomuistista 2, vaan toiminta voi jatkua esim. seuraavan lohkon tyhjennyksellä, jota esittävät nuolet 206 ja 207 sekä lohko 208 kuvassa 2a.

10 Tässä esimerkkitilanteessa on muistiohjain 3 havainnut tilatiedon tarkistuksen 208 yhteydessä, että lohkomuistissa 2 on havaittu virheen tapahtuneen lohkon tyhjennyksessä. Tällöin muistiohjain 3 edullisesti muodostaa keskeytyksen (nuoli 209) suorittimelle 4, joka tämän keskeytyksen perusteella voi ryhtyä tarvittaviin toimenpiteisiin, kuten yrittää lohkon tyhjennystä uudelleen. Mikäli lohkon tyhjentäminen ei mahdollisista useista tyhjennysyrityksistä huolimatta onnistu, voidaan olettaa,

15 että kyseisessä lohkoissa on yksi tai useampi viallinen muistisolun. Tällöin ao. muistilohko voidaan merkitä käyttökelvottomaksi ja siihen ei yritetä myöhemmin kohdistaa muistitoimenpiteitä. On kuitenkin selvää,

20 että joissakin sovelluksissa koko lohko ei välttämättä ole käyttökelvoton, vaan vialliseksi voidaan koko lohkon sijasta merkitä sellainen sivu, jossa viallinen muistisolun sijaitsee. Lisäksi lohkomuistien yhteyteen on mahdollista järjestää virheenkorjaustoimintoja, jolloin osa viallisista muistisolusta on korvattavissa toisella muistisolulla tai viallisen muistisolun vaikutusta voidaan yrittää eliminoida virheenkorjausalgoritmeilla tai vastaavalla sinänsä tunnetulla menetelmällä.

25

Kuvataan seuraavaksi tietojen tallentamista lohkomuistiin 2. Tätä on esitetty kuvassa 2b signaalintikaaviona. Suoritin 4 lähettää muistiohjaimelle 3 sen toimintatilaan vaikuttavia toimintaparametreja (nuoli 211 kuvan 2b kaaviossa). Tämän jälkeen suoritin 4 lähettää muistiohjaimelle 3 muistisivun tallennuskomennon, tallennettavan tiedon sekä osoitteen, johon tieto on tallennettava lohkomuistissa 2 (nuoli 212). Tämä tallennusosoite on sopivimmin sivuosoite. Kuten jo aikaisemmin

30 on todettu, ei lohkomuisteissa yksittäistä tavua voida tallentaa sellaiseen, vaan koko sen sivun tiedot on tallennettava, jossa muutos tapahtuu. Jos tallennettava tietoalue on suurempi kuin yksi sivu, ilmoittaa

35

suoritin 4 muistiohjaimelle edullisesti myös esim. tallennettavien tavujen, sivujen tai lohkojen määrän tai tallennettavan alueen loppuosoitteen. Toisaalta jos suoritin 4 lähettää muistiohjaimelle 3 kaikki tallennettavat tiedot, voi muistiohjain vastaanottamiensa tietojen lukumäärän perusteella itse laskea tallennettavan tiedon määrän sekä sen, kuinka monta sivua tallennettavaa tietoa on.

Tämän jälkeen muistiohjain 3 suorittaa vastaavat toimenpiteet lohkomuistille 2, kuin edellä tyhjennysesimerkissä on esitetty, eli asettaa tarvittaessa piirivalintalinjan 5d tilan, ja tutkii tilalinjan 5h tilan. Tilalinjan 5h ilmaistessa, että lohkomuisti 2 on valmis vastaanottamaan komen-toja, asettaa muistiohjain 3 liitäntäväylän 5 dataväylälle 5a tietojen tallennuskomennon (nuoli 213). Tämän jälkeen komennon liipaisulinjalle 5b lähetetään edullisesti pulssi, jolloin lohkomuistin 2 sisäinen ohjain 16 lukee komennon dataväylältä 5a pulssin reunalla. Seuraavaksi muistiohjain 3 asettaa dataväylälle 5a osoitetiedon tai osan siitä ja sen jälkeen kun osoitetieto on asettunut dataväylälle 5a, muistiohjain 3 muodostaa pulssin osoitteen liipaisulinjaan 5c. Tällöin lohkomuistin ohjain lukee osoitetiedon dataväylältä 5a. Sen jälkeen kun koko osoite on välitetty lohkomuistiin 2, vaihtaa lohkomuisti 2 tilalinjan 5h tilan, eli asettaa sen toiminto kesken –tilaan. Tämän jälkeen lohkomuistissa 2 aloitetaan tietojen tallentamisen muistisoluihin. Kun tilalinjan 5h tila on toiminto kesken –tilassa, on lohkomuisti 2 suorittamassa asetettua toimintoa, eli tässä tapauksessa tietojen tallentamista muistiin.

Myös tietojen tallennuksen yhteydessä lohkomuistissa 2 tarkistetaan, onnistuiko kaikkien bittien asettaminen haluttuun tilaan vai ei. Mikäli jokin bitti ei asettunut haluttuun tilaan, tallennetaan virhetieto lohkomuistissa 2. Tallennus- ja vertailutoiminnon suorittamisen jälkeen tilalinjan 5h tila palautetaan valmiina -tilaan (nuoli 214). Tämä tilalinjan 5h tilan vaihtuminen aiheuttaa muistiohjaimella 3 keskeytyksen, jolloin muistiohjain 3 siirtyy suorittamaan vastaavaa keskeytyspalveluohjelmaa. Jos tallennuksessa tapahtui virhe, välitetään tästä virhetieto vastaavalla tavalla kuin tyhjennyksen yhteydessä edullisesti seuraavasti. Lohkomuistissa 2 vaihdetaan tilalinjan 5h tila takaisin toiminto kesken –tilaan. Nyt muistiohjain 3 lukee tilalinjan 5h tilan (lohko 215), ja jos se on mainitussa toiminto kesken –tilassa, voi muistiohjain 3 päätellä vir-

heen tapahtuneen. Jos tilalinjan 5h tila sen sijaan on valmiina-tilassa, voi muistiohjain 3 päätellä toiminnon onnistuneen ja seuraava komento voidaan välittää lohkomuistiin 2. Jos tiedot tallentuivat oikein, voidaan tietojen tallennusta jatkaa tarvittaessa. Tällöin muistiohjain 3 lähettää

5 tallennuskomennon, edullisesti seuraavan sivun osoitteen sekä kyseiselle sivulle tallennettavat tiedot (nuoli 216). Keksinnön edullisen suoritusmuodon mukaisessa menetelmässä muistiohjain 3 suorittaa osoitteiden kasvattamisen, jolloin suorittimen 4 ei tarvitse huolehtia siitä. Sen jälkeen kun sivun tiedot on tallennettu lohkomuistiin 2, muodostetaan jälleen keskeytys muistiohjaimelle 3 (nuoli 217), joka tämän jälkeen käy tarkistamassa tilalinjan tilan (lohko 218). Jos on havaittu tallennusvirhe, muistiohjain 3 edullisesti muodostaa keskeytyksen (nuoli 219) suorittimelle 4, joka tämän keskeytyksen perusteella voi ryhtyä tarvittaviin toimenpiteisiin, kuten yrittää tietojen tallennusta uudelleen.

15 Joissakin muisteissa voidaan tietoa myös kopioida muistin sisällä yhdestä lohkosta johonkin toiseen lohkoon ilman, että kopioitavaa tietoa tarvitsee siirtää muistin ulkopuolelle. Tästä toiminnosta käytetään joissakin yhteyksissä englanninkielistä nimitystä Copy-Back tekniikka. Kopiointitoiminnon suorittamiseksi edullisesti suoritin 4 lähettää muistiohjaimelle 3 muistisivun kopiointikomennon, tietoa kopioitavan tiedon sijainnista muistissa (esim. alkuosoite), sekä osoitteen, johon tieto on tarkoitus kopioida lohkomuistissa 2 (esim. ao. sivun sivuosoitteen). Jos kopioitava tietoa alue on suurempi kuin yksi sivu, ilmoittaa suoritin 4

20 muistiohjaimelle edullisesti myös esim. kopioitavien tavujen, sivujen tai lohkojen määrän, tai kopioitavan alueen loppuosoitteen. Tarvittavien tietojen vastaanottamisen jälkeen muistiohjain 3 välittää tiedot lohkomuistiin 3, jossa kopiointi voidaan aloittaa. Komentojen välityksessä noudatetaan edullisesti edellä tallennuksen yhteydessä esitettyjä toimintaperiaatteita. Aloittaessaan tietojen kopiointin lohkomuisti 3 asettaa tilalinjan 5h tilan toiminto kesken –tilaan. Tämän jälkeen lohkomuistissa 2 aloitetaan tietojen kopiointi. Kun tilalinjan 5h tila on toiminto kesken –tilassa, on lohkomuisti 2 suorittamassa tietojen kopiointia muistissa. Tietojen kopiointin yhteydessä lohkomuistissa 2 tarkistetaan, onnistuiko kopiointi vai ei. Mikäli kopiointissa tapahtui virhe, tallennetaan virhetieto lohkomuistissa 2. Kopiointi- ja vertailutoiminnon suorittamisen jälkeen tilalinjan 5h tila palautetaan valmiina -tilaan. Jos

kopioinnissa tapahtui virhe, vaihdetaan tilalinjan 5h tila takaisin toiminto kesken –tilaan, jolloin muistiohjain 3 ja/tai suoritin 4 voi havaita virheen tapahtuneen.

- 5 Keksintöä voidaan soveltaa myös tietojen lukemisen yhteydessä. Täl-
löinkin lohkomuistiin 2 välitetään lukemisessa tarvittavat tiedot ja ko-
mento/komennot, minkä jälkeen aloitetaan tietojen lukeminen lohko-
muistista 2. Jos lukemisen yhteydessä lohkomuisti 2 havaitsee virheen,
10 suoritetaan siitä ilmoittaminen edellä esitettyjä periaatteita soveltaen
tilalinjan 5h tilamuutosten avulla. Virheen havaitseminen voi perustua
esimerkiksi lohkomuistissa 2 mahdollisesti oleviin sisäisiin tarkkailu-
ominaisuuksiin, joilla solujen toimintakuntoa voidaan tutkia.

- 15 Kuvassa 3 on esitetty keksinnön erään edullisen suoritusmuodon mu-
kaisen lohkomuistin 2 rakennetta pelkistetysti. Lohkomuisti käsittää
ohjaimen 16, joka suorittaa suurelta osin edellä kuvattujen toimintojen
ohjaamisen, kuten statustietojen ilmaisun, tietojen vertailun, tyhjennyk-
sen/tallennuksen ohjaamisen lohkomuistissa 2, yms. Lisäksi lohko-
muistissa 2 on I/O-lohko 17 lohkomuistin 2 liittämiseksi liittäntävyylään
20 5, muistisolumatriisi 18, rivi- ja sarakeosoitepuskurit 19, 20, datapuskuri
21 sekä rekistereitä 22.

- 25 Kuvan 1 mukaisessa elektroniikkalaitteessa on myös DMA-lohko 8
(Direct Memory Access). Tämän DMA-lohkon 8 avulla voidaan suorit-
taa tietojen siirtoa elektroniikkalaitteen muistin 9 ja muistiohjaimen 3
välillä ilman, että suorittimen 4 tarvitsee osallistua tietojen siirtoon.
Tällöin suoritin 4 voi tietojen siirron aikana suorittaa muita tehtäviä.
Tällöin nyt esillä olevaa keksintöä voidaan soveltaa myös siten, että
suoritin 4 käynnistää tietojen siirtämisen esim. muistista 9 lohkomuistiin
30 2 edellä esitettyjä periaatteita soveltaen. Muistiohjain 3 huolehtii tieto-
jen hakemisesta DMA-lohkon 8 avulla muistista 9, haettujen tietojen
tallentamisesta lohkomuistiin ja tallennettujen tietojen oikeellisuuden
varmistamisesta. Virhetilanteessa muistiohjain 3 voi esim. keskeytyk-
sellä ilmoittaa suorittimelle 4 virheestä, jolloin suoritin 4 voi ryhtyä vir-
35 heen aiheuttamiin toimenpiteisiin. Toisaalta keksintöä voidaan soveltaa
myös siten, että muistiohjain 3 yrittää ensin poistaa virheellisesti tallen-

netun tiedon esim. yrittämällä tallennusta uudelleen, ja jos se ei onnistu, muistiohjain 3 ilmoittaa virheestä suorittimelle 4.

5 Vaikka edellä keksintöä on kuvattu elektroniikkalaitteessa, jossa käytetään muistiohjainta 3 ja mahdollisesti myös DMA-lohkoa 8, on selvää, että keksintöä voidaan soveltaa myös ilman muistiohjainta 3 ja DMA-lohkoa 8. Jos muistiohjainta 3 ei käytetä, voidaan lohkomuisti 2 liittää esim. suorittimen 4 väylään. Tällöin suorittimessa 4 suoritetaan edellä muistiohjaimen 3 suorittamat toimenpiteet. Myös tässä tapauksessa nyt
10 esillä olevalla keksinnöllä voidaan suorittimen 4 kuormitusta vähentää mm. siitä syystä, että tilatietoja ei tarvitse erikseen käydä lukemassa lohkomuistista 2, vaan ne selviävät tutkimalla lohkomuistin tilalinjan 5h tila.

15 Lohkomuistin 2 tilalinjana ei välttämättä tarvitse käyttää edellä esitettyä tilalinjaa 5h, vaan sen tilalla voidaan käyttää muutakin liitäntäväylän 5 linjaa, kuten jotakin dataväylän 5a datalinjaa, tai tähän tarkoitukseen voidaan järjestää oma linjansa.

20 Mainittakoon tässä yhteydessä vielä se, että keskeytysten tilalla voidaan käyttää myös muutakin menetelmää, jolla lohkomuistin toimintatilaa voidaan seurata tietojen tallennuksen ja tyhjennyksen aikana. Tällöin esim. muistiohjain 3 käy väliajoin lukemassa tilalinjan 5h tilan ja kun muistiohjain havaitsee tilan muuttuneen toiminto kesken –tilasta
25 valmiina-tilaan, voidaan tästä päätellä se, että lohkomuisti 2 on suorittanut edellisellä komennolla aloitetun toiminnon loppuun. Tämän jälkeen muistiohjain 3 voi tarkkailla, vaihtaako lohkomuisti 2 tämän tilalinjan tilan suhteellisen pian takaisin toiminto kesken –tilaksi, mikä siis on merkinä siitä, että tyhjennys/tallennusvirhe on havaittu lohkomuistissa 2.
30

Virhetilanteen jälkeen lohkomuisti 2 asettaa tilalinjan tilan takaisin valmiina-tilaan edullisesti viimeistään siinä vaiheessa kun lohkomuistiin kohdistetaan seuraava operaatio. Lohkomuisti voi havaita tämän jonkin
35 ohjauslinjan tilamuutoksesta, kuten piirinvalintalinjan 5d tilanmuutoksesta.

Kuvan 1 mukaisessa elektroniikkalaitteessa 1 on myös käyttöliittymä, jossa on mm. näyttö 10, näppäimistö 11, kuuloke 12 ja mikrofoni 13. Lisäksi elektroniikkalaitteessa 1 on vielä lähetin 14 ja vastaanotin 15 matkaviestintoimintojen suorittamiseksi.

5

Keksintöä voidaan soveltaa myös muistikorttien (ei esitetty) yhteydessä. Tällaisia muistikortteja voidaan liittää elektroniikkalaitteeseen 1 järjestetyn korttiliitännän avulla. Tällöin edellä esitetyn kaltainen komentojen ja signaalien välitys suoritetaan korttiliitännän kautta. Muistikortille on edullisesti järjestetty keksinnön mukaista menetelmää soveltavaa lohkomuistia.

10

On selvää, että nyt esillä olevaa keksintöä ei ole rajoitettu ainoastaan edellä esitettyihin suoritusmuotoihin, vaan sitä voidaan muunnella oheisten patenttivaatimusten puitteissa.

15

Patenttivaatimukset:

1. Menetelmä muistin käsittelytoiminnon suorittamiseksi lohkomuistissa (2), jossa on muistisoluja (18), joihin tietoa voidaan tallentaa, ja väyläliitäntä (5), jossa on ainakin tilalinja (5h), joka on asetettavissa ainakin valmiina-tilaan ja toiminto kesken -tilaan, ja jossa menetelmässä suoritetaan käsittelytoiminnon alussa mainitun tilalinjan (5h) tilan asettaminen mainittuun toiminto kesken -tilaan, ja suoritetaan käsittelyvirheiden havainnointi, ja käsittelytoiminnon päätyminen ilmaistaan asettamalla mainittu tilalinja (5h) mainittuun valmiina-tilaan, **tunnettu** siitä, että mikäli havaitaan käsittelyvirhe, mainitun tilalinjan tila muutetaan takaisin mainittuun toiminto kesken -tilaan.
2. Patenttivaatimuksen 1 mukainen menetelmä, **tunnettu** siitä, että mainittu käsittelytoiminto on lohkomuistin (2) tyhjentäminen, jolloin tyhjennyksen yhteydessä suoritetaan tyhjennetyin muistisolun tiedon vertaaminen tyhjän muistisolun tilaksi tarkoitettuun tietoon tyhjennysvirheiden havaitsemiseksi, ja mikäli vertailussa havaitaan tyhjennysvirhe, mainitun tilalinjan tila muutetaan takaisin mainittuun toiminto kesken -tilaan.
3. Patenttivaatimuksen 1 mukainen menetelmä, **tunnettu** siitä, että mainittu käsittelytoiminto on tietojen tallentaminen lohkomuistiin (2), jolloin tietojen tallennuksen yhteydessä suoritetaan tallennetun tiedon vertaaminen tallennettavaksi tarkoitettuun tietoon tallennusvirheiden havaitsemiseksi, ja mikäli vertailussa havaitaan tallennusvirhe, mainitun tilalinjan tila muutetaan takaisin mainittuun toiminto kesken -tilaan.
4. Patenttivaatimuksen 1 mukainen menetelmä, **tunnettu** siitä, että mainittu käsittelytoiminto on tietojen kopiointi lohkomuistissa (2) yhdestä paikasta johonkin toiseen paikkaan, jolloin tietojen kopiointin yhteydessä suoritetaan kopioitun tiedon vertaaminen kopiotavaan tietoon kopiointivirheiden havaitsemiseksi, ja mikäli vertailussa havaitaan kopiointivirhe, mainitun tilalinjan tila muutetaan takaisin mainittuun toiminto kesken -tilaan.

5. Patenttivaatimuksen 1 mukainen menetelmä, **tunnettu** siitä, että mainittu käsittelytoiminto on tietojen lukeminen lohkomuistista (2), jolloin tietojen lukemisen yhteydessä suoritetaan luettavana olevan muistisolun tilan tutkiminen, ja mikäli tutkimisessa havaitaan muistisoluvirhe, mainitun tilalinjan tila muutetaan takaisin mainittuun toiminto kesken –tilaan.
6. Jonkin patenttivaatimuksen 1—5 mukainen menetelmä, **tunnettu** siitä, että käsittelytoiminto käynnistetään lähettämällä komento lohkomuistiin (2), ja että mainitun tilalinjan (5h) tila asetetaan toiminto kesken –tilaan siinä vaiheessa kun komennon mukainen toiminto käynnistetään lohkomuistissa (2).
7. Jonkin patenttivaatimuksen 1—6 mukainen menetelmä, **tunnettu** siitä, että lohkomuisti (2) on jaettu lohkoihin, ja kukin lohko on jaettu sivuihin, jolloin tietoa välitetään lohkomuistiin (2) sivuittain.
8. Patenttivaatimuksen 7 mukainen menetelmä, **tunnettu** siitä, että suoritetaan ainakin yhden lohkon tyhjennys, jolloin kaikkien mainitun ainakin yhden lohkon muistisolujen tilaksi asetetaan ennalta määrätty tila.
9. Jonkin patenttivaatimuksen 1—8 mukainen menetelmä, **tunnettu** siitä, että tutkitaan mainitun tilalinjan (5h) tilaa, jolloin sen muuttuessa toiminto kesken –tilasta valmiina-tilaan, suoritetaan tilalinjan (5h) tilan uudelleen tutkiminen.
10. Jonkin patenttivaatimuksen 1—9 mukainen menetelmä, **tunnettu** siitä, että väyläliitännässä (5) on ainakin yhden datalinjan käsittävä dataväylä (5a), ja että tilalinjana (5h) käytetään mainitun dataväylän (5a) datalinjaa.
11. Järjestelmä, joka käsittää elektroniikkalaitteen (1), jossa on lohkomuistia (2), jossa on muistisoluja (18), joihin tietoa on järjestetty tallennettavaksi, ja väyläliitäntä (5), jossa on ainakin tilalinja (5h), joka on järjestetty asetettavaksi ainakin valmiina-tilaan ja toiminto kesken -tilaan, ja joka lohkomuisti (2) käsittää välineet (16) mainitun tilalin-

5 jan (5h) tilan asettamiseksi mainittuun toiminto kesken –tilaan lohkomuistin käsittelytoiminnon alussa, vertailijan (16) käsittelyvirheiden havaitsemiseksi, ja käsittelytoiminnon päätyminen on järjestetty ilmaistavaksi asettamalla mainittu tilalinja (5h) mainittuun valmiina-tilaan, **tunnettu** siitä, että käsittelyvirheen havaitsemisen jälkeen mainitun tilalinjan (5h) tila on järjestetty muutettavaksi takaisin mainittuun toiminto kesken –tilaan.

10 12. Patenttivaatimuksen 11 mukainen järjestelmä, **tunnettu** siitä, että mainittu käsittelytoiminto on jokin seuraavista:

- muistisolujen (18) tyhjentäminen,
- tietojen tallentaminen muistisoluihin (18),
- tietojen kopiointi muistisolujen (18) välillä,
- muistisoluihin (18) tallennetun tiedon lukeminen.

15 13. Patenttivaatimuksen 11 tai 12 mukainen järjestelmä, **tunnettu** siitä, että se käsittää suorittimen (4), ja ensimmäisen väyläliitännän suorittimen (4) ja lohkomuistin (2) välillä, jolloin käsittelytoiminto on järjestetty käynnistettäväksi lähettämällä komento lohkomuistiin (2), ja
20 että mainitun tilalinjan (5h) tila on järjestetty asetettavaksi toiminto kesken –tilaan siinä vaiheessa kun komennon mukainen toiminto on käynnistetty lohkomuistissa (2).

25 14. Patenttivaatimuksen 13 mukainen järjestelmä, **tunnettu** siitä, että se käsittää välineet (3, 5, 7) keskeytyksen aikaansaamiseksi suorittimessa (4) tilalinjan (5h) tilan muuttuessa toiminto kesken –tilasta valmiina-tilaan, jolloin suorittimessa (4) on järjestetty tutkittavaksi tilalinjan (5h) tila keskeytyksen käsittelyn yhteydessä.

30 15. Patenttivaatimuksen 11 tai 12 mukainen järjestelmä, **tunnettu** siitä, että se käsittää suorittimen (4), muistiohjaimen (3), ensimmäisen väyläliitännän suorittimen (4) ja muistiohjaimen (3) välillä, ja toisen väyläliitännän muistiohjaimen (3) ja lohkomuistin (2) välillä, jolloin komentoja on järjestetty lähetettäväksi suorittimesta (4) muistiohjaimeen,
35 sekä muistiohjaimesta lohkomuistiin (2), jolloin tietojen tallennus on järjestetty käynnistettäväksi lähettämällä komento lohkomuistiin (2), ja että mainitun tilalinjan (5h) tila on järjestetty asetettavaksi toiminto kes-

ken –tilaan siinä vaiheessa kun komennon mukainen toiminto on käynnistetty lohkomuistissa (2).

5 16. Patenttivaatimuksen 15 mukainen järjestelmä, **tunnettu** siitä, että se käsittää välineet (5) keskeytyksen aikaansaamiseksi muistiohjaimessa (3) tilalinjan (5h) tilan muuttuessa toiminto kesken –tilasta valmiina-tilaan, jolloin muistiohjaimessa on järjestetty tutkittavaksi tilalinjan (5h) tila keskeytyksen käsittelyn yhteydessä, ja mikäli tilalinja on toiminto kesken –tilassa, muistiohjaimesta (3) on järjestetty muodostettavaksi keskeytys suorittimelle (4).

15 17. Jonkin patenttivaatimuksen 11—16 mukainen järjestelmä, **tunnettu** siitä, että lohkomuisti (2) on jaettu lohkoihin, ja kukin lohko on jaettu sivuihin, jolloin tietoa on järjestetty välitettäväksi lohkomuistiin (2) sivuittain.

20 18. Jonkin patenttivaatimuksen 11—17 mukainen järjestelmä, **tunnettu** siitä, että väyläliitännässä (5) on ainakin yhden datalinjan käsittävä dataväylä (5a), ja että tilalinjana (5h) on järjestetty käytettäväksi mainitun dataväylän (5a) datalinjaa.

25 19. Elektroniikkalaite (1), joka käsittää lohkomuistia (2), jossa on muistisoluja (18), joihin tietoa on järjestetty tallennettavaksi, ja väyläliitântä (5), jossa on ainakin tilalinja (5h), joka on järjestetty asetettavaksi ainakin valmiina-tilaan ja toiminto kesken -tilaan, ja joka lohkomuisti (2) käsittää välineet (16) mainitun tilalinjan (5h) tilan asettamiseksi mainittuun toiminto kesken –tilaan lohkomuistin käsittelytoiminnon alussa, ja vertailijankäsittelyvirheiden havaitsemiseksi, ja käsittelytoiminnon päättymisen on järjestetty ilmaistavaksi asettamalla mainittu tilalinja (5h) mainittuun valmiina-tilaan, **tunnettu** siitä, että käsittelyvirheen havaitsemisen jälkeen mainitun tilalinjan (5h) tila on järjestetty muutettavaksi takaisin mainittuun toiminto kesken –tilaan.

35 20. Lohkomuisti (2), jossa on muistisoluja (18), joihin tietoa on järjestetty tallennettavaksi, ja väyläliitântä (5), jossa on ainakin tilalinja (5h), joka on järjestetty asetettavaksi ainakin valmiina-tilaan ja toiminto kesken -tilaan, ja joka lohkomuisti (2) käsittää välineet (16) mainitun tila-

linjan (5h) tilan asettamiseksi mainittuun toiminto kesken –tilaan lohkomuistin käsittelytoiminnon alussa, ja vertailijankäsittelyvirheiden havaitsemiseksi, ja käsittelytoiminnon päätyminen on järjestetty ilmaista-
vaksi asettamalla mainittu tilalinja (5h) mainittuun valmiina-tilaan,
5 **tunnettu** siitä, että käsittelyvirheen havaitsemisen jälkeen mainitun tilalinjan (5h) tila on järjestetty muutettavaksi takaisin mainittuun toiminto kesken –tilaan.

21. Patenttivaatimuksen 20 mukainen lohkomuisti (2), **tunnettu** siitä,
10 että mainittu käsittelytoiminto on jokin seuraavista:

- muistisolujen (18) tyhjentäminen,
- tietojen tallentaminen muistisoluihin (18),
- tietojen kopiointi muistisolujen (18) välillä,
- muistisoluihin (18) tallennetun tiedon lukeminen.

22. Patenttivaatimuksen 20 tai 21 mukainen lohkomuisti (2),
15 **tunnettu** siitä, että se käsittää vertailijan (16) muistisoluihin (18) tallennetun tiedon vertaamiseksi tallennettavaksi tarkoitettuun tietoon tallennusvirheiden havaitsemiseksi.

(57) Tiivistelmä:

Keksintö kohdistuu menetelmään muistin käsittelytoiminnon suorittamiseksi lohkomuistiin (2), jossa on muistisoluja (18), joihin tietoa voidaan tallentaa, ja väylälitettä (5), jossa on ainakin tilalinja (5h). Tilalinja (5h) on asetettavissa ainakin valmiina-tilaan ja toiminto kesken – tilaan. Menetelmässä suoritetaan käsittelytoiminnon alussa mainitun tilalinjan (5h) tilan asettaminen mainittuun toiminto kesken –tilaan, ja suoritetaan käsittelyvirheiden havainnointi. Käsittelytoiminnon päätyminen ilmaistaan asettamalla mainittu tilalinja (5h) mainittuun valmiina-tilaan. Käsittelyvirheen havaitsemisen jälkeen mainitun tilalinjan (5h) tila muutetaan takaisin mainittuun toiminto kesken –tilaan, mikäli havaitaan käsittelyvirhe. Keksintö kohdistuu lisäksi järjestelmään, elektroniikkalaitteeseen (1) ja lohkomuistiin (2), joissa menetelmää sovelletaan.

Fig. 2a

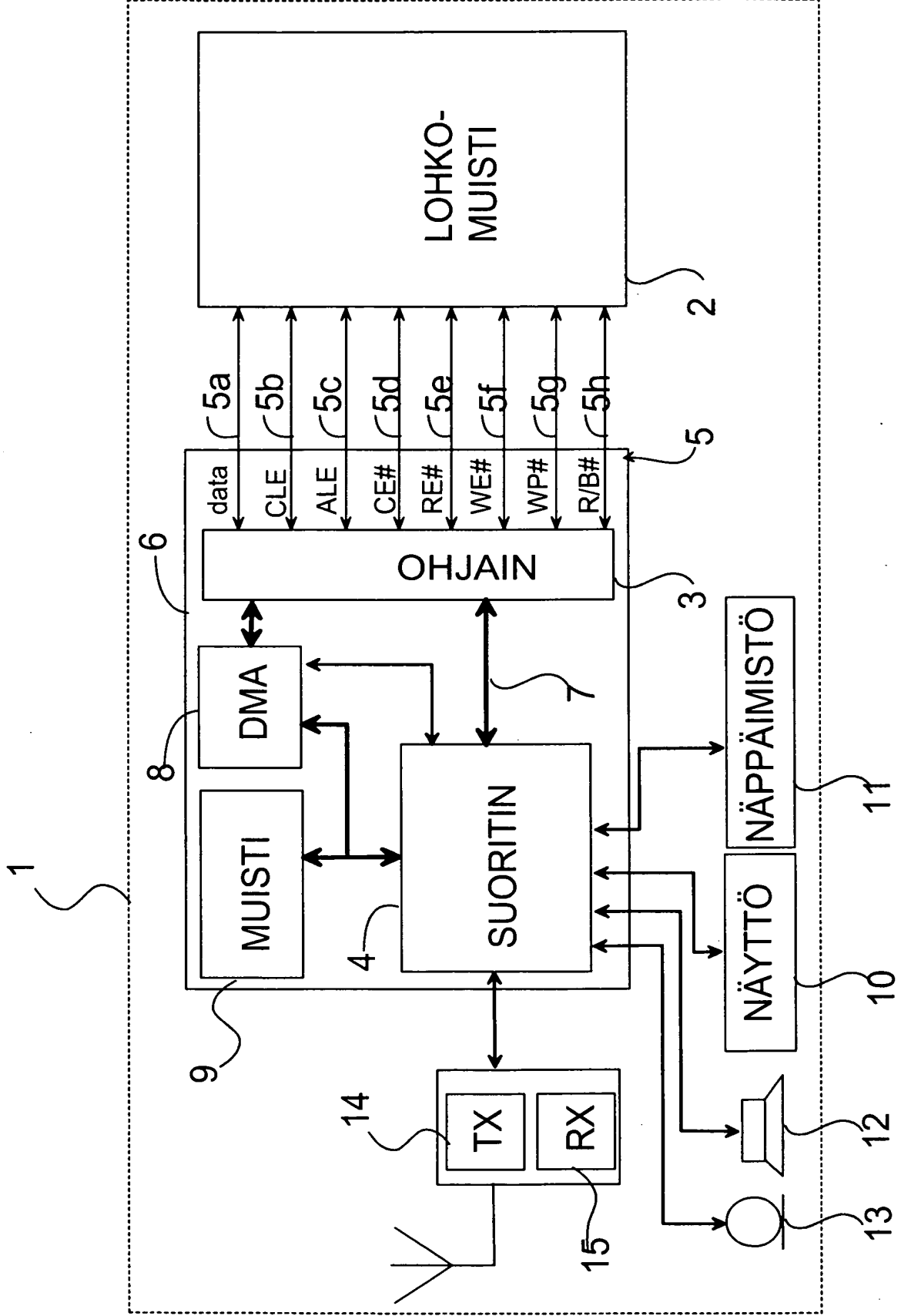


Fig. 1

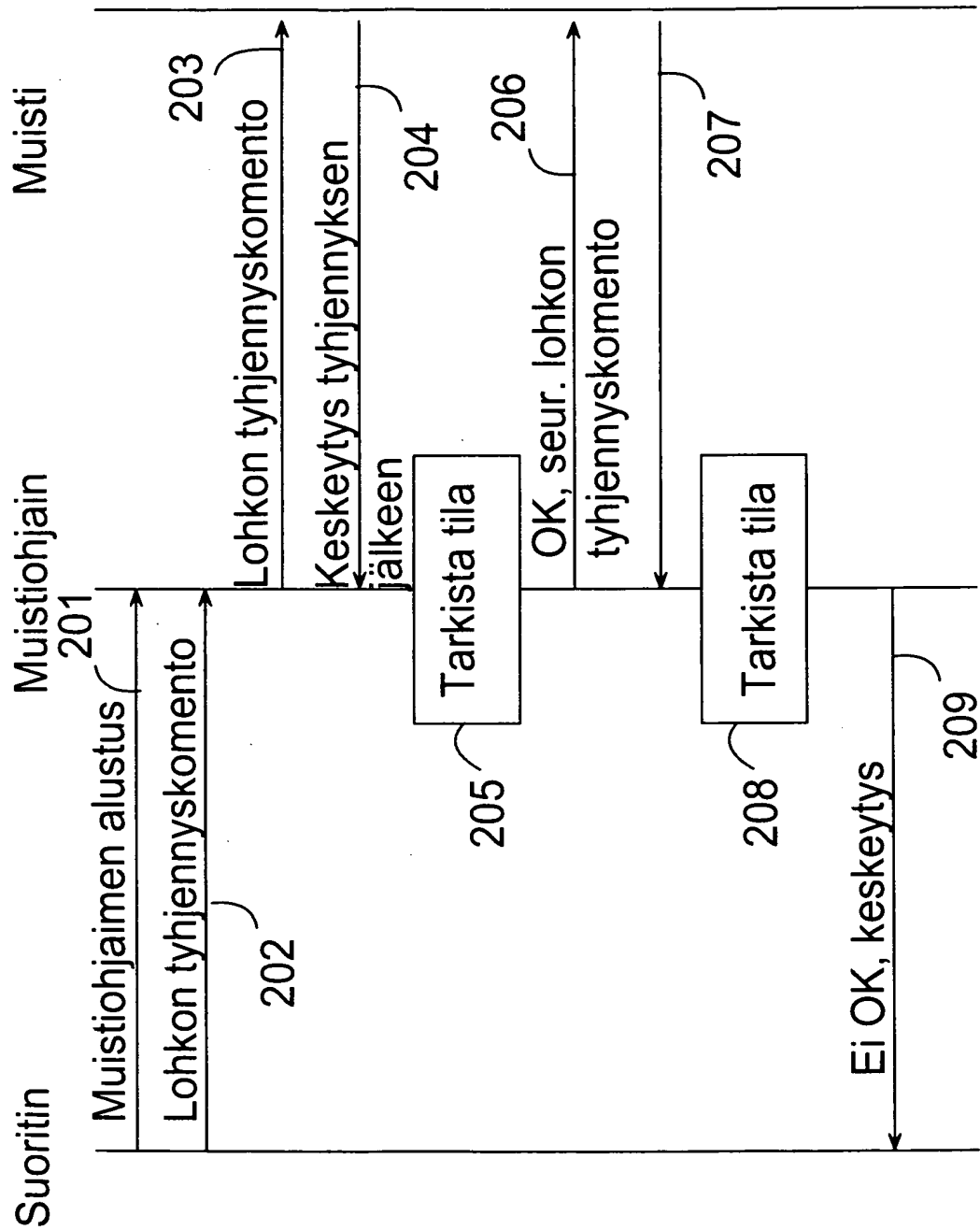


Fig. 2a

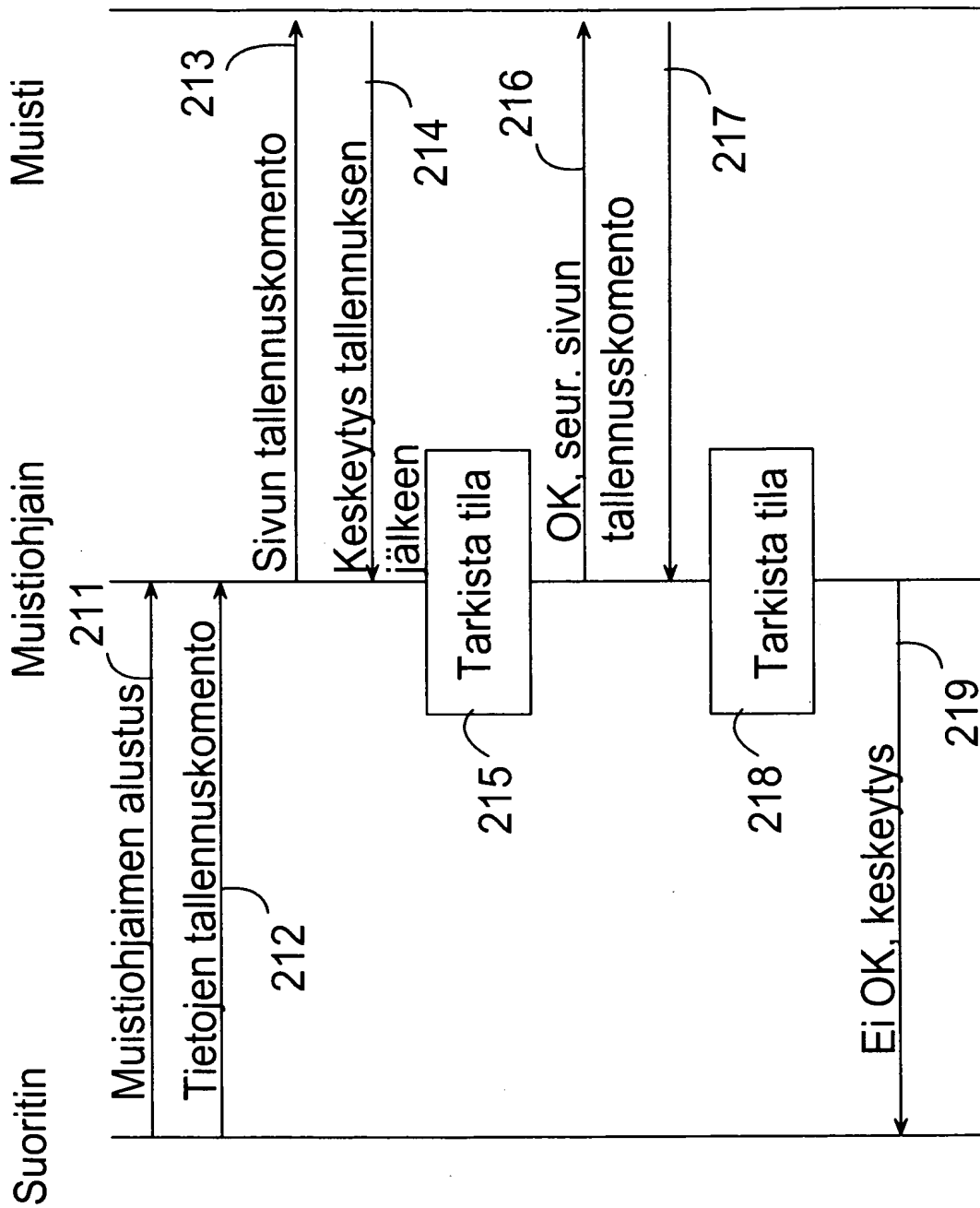


Fig. 2b

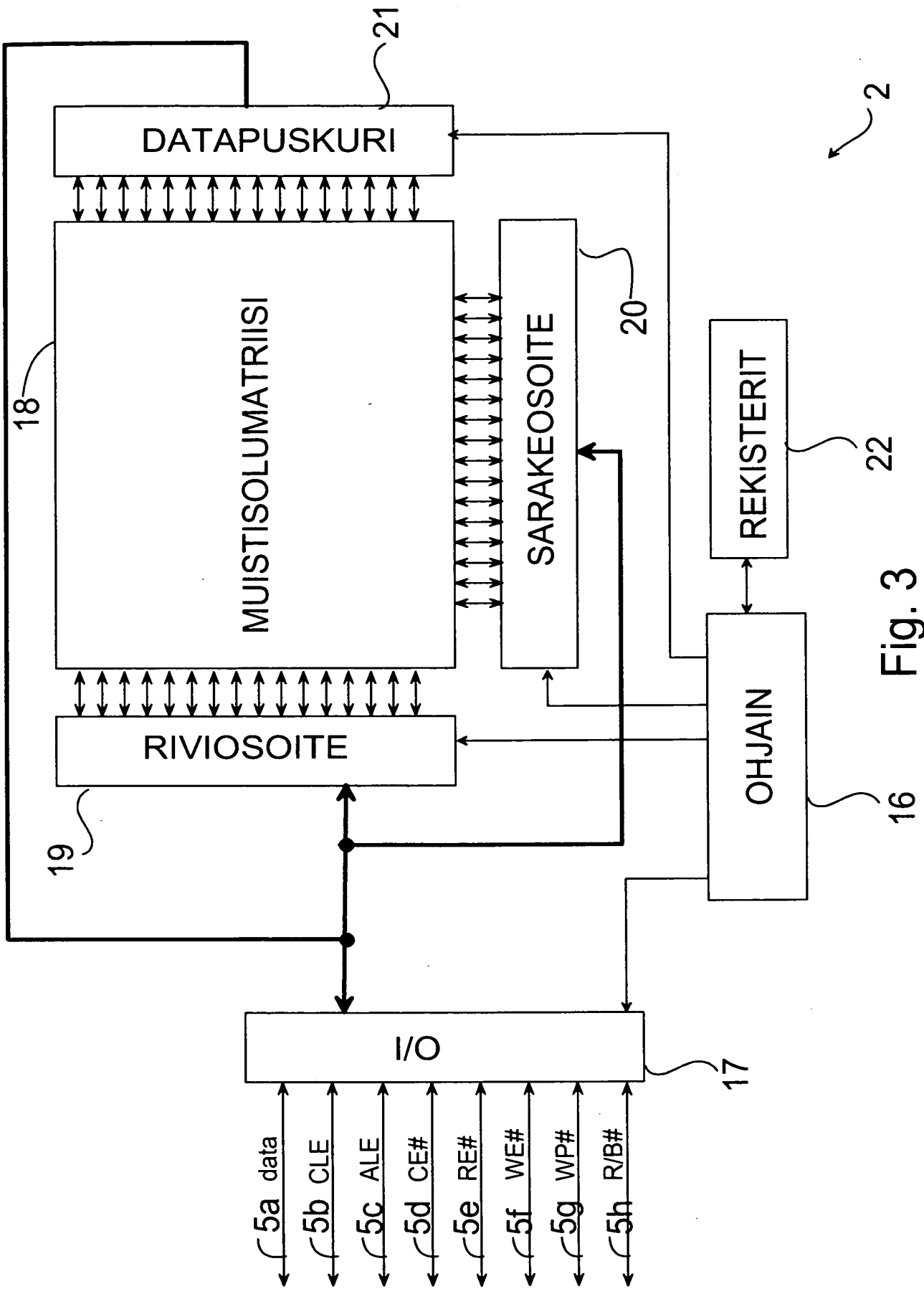


Fig. 3

CERTIFICATE

I, Tuulikki Tulivirta, hereby certify that, to the best of my knowledge and belief, the following is a true translation, for which I accept responsibility, of Finnish Patent Application No. 20035041 filed on 31 March 2003.

Tampere, 10 March 2004



Tuulikki Tulivirta
Certified Translator (Act 1148/88)

Tampereen Patenttitoimisto Oy
Hermiankatu 12 B
FIN-33720 TAMPERE
Finland

A method for storing data in a memory, a system, an electronic device and a memory card

5 The present invention relates to a method for performing a memory processing function in a block memory, wherein there are memory cells, where data can be stored, and a connection bus, wherein there is at least a ready/busy line, which can be set at least to a ready status and to a busy status, and in which method setting the status of said
10 ready/busy line to said busy status is performed in the beginning of the processing function, and detection of processing errors is performed, and the end of processing function is indicated by setting said ready/busy line into said ready status. In addition, the present invention relates to a system, which comprises an electronic device, wherein
15 there is block memory, where there are memory cells, wherein data is arranged to be stored, and a connection bus, wherein there is at least a ready/busy line, which is arranged to be set at least to a ready status and to a busy status, and which block memory comprises means for setting the status of said ready/busy line to said busy status in the
20 beginning of the processing function of the block memory, a comparator for detecting processing errors, and the end of processing function is arranged to be indicated by setting said ready/busy line into said ready status. In addition, the present invention relates to an electronic device, which comprises block memory, wherein there are
25 memory cells, where data is arranged to be stored, and a connection bus, wherein there is at least a ready/busy line, which is arranged to be set at least to a ready status and to an busy status, and which block memory comprises means for setting the status of said ready/busy line to said busy status in the beginning of the processing function of the
30 block memory, a comparator for detecting processing errors, and the end of processing function is arranged to be indicated by setting said ready/busy line into said ready status. The invention also relates to a block memory, wherein there are memory cells, where data is arranged to be stored, and a connection bus, wherein there is at least a
35 ready/busy line, which is arranged to be set at least to a ready status and to a busy status, and which block memory comprises means for setting the status of said ready/busy line to said busy status in the

beginning of the processing function of the block memory, a comparator for detecting processing errors, and the end of processing function is arranged to be indicated by setting said ready/busy line into said ready status.

5

There are known memories, where data programming into the memory and reading the stored data is performed in larger units instead of bytes/words. For example, NAND memories and NOR memories consist of memory cell matrixes. Each line of this type of a matrix forms one page. Larger assemblies (blocks) can be gathered from these lines, in which case a part of memory operations is page-specific and a part relates to one or more blocks. The size of a page is, for example, in a 64-megabyte NAND memory 528 bytes, from which 16 bytes are so called Spare bytes, i.e. the number of usable bytes on one page is 512. In this example memory, one block consists of 16 pages, but other block sizes can also be used. In this description, such memories are also called block memories.

NAND memories do not include an address bus, but communicating the addresses to the memory is performed via a data bus. The width of the data bus is typically 8 bits. Typically, in NAND memories the programming of data into the memory and reading the data from the memory is performed one page at a time, but, for example, erasing the memory (setting the memory cell to a certain value, either to a 0-status or to a 1-status) is performed by blocks. Changing the data of an individual byte requires rewriting the data of that entire page where the byte to be changed is located. The use of this type of memory is somewhat more complicated than the use of a conventional Random Access Memory (RAM). When data is written to this type of memory, the operation is typically such, that the processor of a device sends the data to be stored one page at a time to the memory controller, which forwards the data to the memory. The memory is first informed the address of the programming location, typically the base address of the page. This takes place in such a manner, that a command referring to this is set to the I/O bus, after which a pulse is set to the command line, in which case the memory reads the command in the I/O bus. In addition to the command, also the address of the programming location

is provided this way. In the next step, sending the data from the pages to be saved from the memory controller to the memory is started, typically bit by bit. There is cache in the memory, where the page to be stored into the memory is written.

5

After storing the data, the validity of programming is checked. In some memories according to prior art this is performed automatically after each stored page by comparing the stored data to the data that was to be stored. The status of the comparison is stored into a status register inside the memory. The status of this status register can be checked after the programming by sending a command to read the status register to the memory, which the memory answers with the contents of the status register. Thus, the memory controller can, on the basis of this status data, conclude whether the data was stored correctly. In order to send the status data of programming, the memory controller communicates, for example, an interrupt request to the processor, which on the basis of this reads the status data sent by the memory controller. This type of an arrangement has e.g. the disadvantage that interruptions come to the processor after the processing of each page, even though the data would be stored in the memory correctly. The amount of data to be stored can be quite large, in which case there are many interruptions, which slows down the operation of the processor.

20

There are several memory structures as described above, which are based on different techniques, such as NAND, XOR, etc. However, what these memories have in common is that the data remains in them after the operating voltages break, i.e. they are so-called non-volatile memories. In addition, the programming of data to this type of non-volatile memories is slower than to dynamic and static random access memories.

30

Non-volatile memories can be arranged as memory cards, which can be attached to an electronic device in a releasable manner. This type of memory cards can thus be moved from one device to another, which makes it possible to e.g. transfer data between devices. For example, in a digital camera the images can be stored into a memory card. After programming the memory card can be removed from the digital camera

35

and placed, for example, in a computer or a mobile communication device. Thus the images can be read from the memory card into the memory of this other device. Other memory card applications include the programming of program files, videos, music, text files, etc.

5

Even though it has been presented above that a memory controller has been arranged for memory processing, via which the transmission of data between the processor and memory is performed, it is clear that the processor can control the memory directly as well. Thus, however, the load in the processor caused by the memory processing is even greater than when using a memory controller.

10

It is an aim of the present invention to provide an improved method and a system for storing data in a memory, as well as a memory and a device applying the method. The invention is based on the idea that information on the programming is transmitted to the processor only if an error has occurred in the programming. This data transmission can be arranged advantageously with an interruption, which can be transmitted along some line existing on the interface of the memory or a line formed for this purpose. To be more exact, the method according to the present invention is primarily characterized in that if a processing error is detected, the status of said ready/busy line is changed back into said busy status. The system according to the present invention is primarily characterized in that after detecting a processing error the status of said ready/busy line is arranged to be changed back into said busy status. The electronic device according to the present invention is primarily characterized in that after detecting a processing error the status of said ready/busy line is arranged to be changed back into said busy status. The block memory according to the present invention is primarily characterized in that after detecting a processing error the status of said ready/busy line is arranged to be changed back into said busy status.

15

20

25

30

35

In a system and an electronic device according to an advantageous embodiment of the invention, the memory controller increases the memory addresses if data is stored consecutively to more than one page.

The present invention shows remarkable advantages over solutions of prior art. With the method according to the present invention, it is possible to decrease the load of the processor, because no interruption is caused to the processor in situations where programming is performed faultlessly. In addition, the need to transmit commands between the memory controller and the memory is smaller than when using memory structures according to prior art. In addition, with the method according to an advantageous embodiment of the invention is reached the advantage that the processor does not have to take care of increasing the memory addresses, in which case the execution time of the processor is freed for performing other tasks.

In the following, the invention will be described in more detail with reference to the appended drawings, in which

- Fig. 1 shows an electronic device according to a preferred embodiment of the invention in a reduced chart,
- Fig. 2a shows the signalling between a processor, a memory controller and a memory, in connection with erasing the memory to be used, in the method according to a preferred embodiment of the invention,
- Fig. 2b shows the signalling between a processor, a memory controller and a memory, in connection with programming to the memory to be used, in the method according to a preferred embodiment of the invention, and
- Fig. 3 shows a block memory according to a preferred embodiment of the invention in a reduced chart.

In the following description, the method according to an advantageous embodiment of the invention will be exemplified with an electronic device 1 shown in Fig. 1, wherein there is, among other things, block memory 1, such as NAND memory or NOR memory. In order to control the block memory 2, a memory controller 3 is advantageously arranged

to the electronic device 1. It is, however, obvious that the memory controller 3 is not necessary in all applications, but, for example, the processor 4 can be used in memory control operations. In this embodiment, the memory controller 3 and the processor 4 are
5 arranged into the same Application Specific Integrated Circuit (ASIC), but these can also be separate circuits.

The connection bus 5 between the block memory 2 and the memory controller 3 comprises advantageously a data bus 5a (data) composed
10 of at least one data line, as well as control lines 5b to 5h. By means of these control lines the performance of memory functions can be controlled. In an electronic device according to this advantageous embodiment, the control lines comprise a Command Latch Enable line 5b (CLE), an Address Latch Enable line 5c (ALE), a Chip Enable line
15 5d (CE), a Read Enable line 5e (RE), a Write Enable line 5f (WE), a Write Protect line 5g (WP), and a Ready/Busy line 5h (R/B). In Fig. 1 the mark # in connection with some control lines means that the line in question is a so-called "active when down" line, i.e. active in a logic 0-status. However, it will be obvious that the invention is not limited solely
20 to alternatives according to Fig. 1, but lines that are active when up (in logic 1-status) can be used.

For transmission of signals, there is a connection bus 7 between the processor 4 and the memory controller 3 as well. Among other things,
25 addresses, data and control data, such as interruptions, are transmitted via this connection bus between the processor 4 and the memory controller 3.

In the method according to a preferred embodiment of the invention,
30 control of the block memory 2 is advantageously performed in the following manner. First, the block-by-block emptying of the block memory 2 is described. This is presented in Fig. 2a as a signalling chart. Processor 4 sends function parameters affecting its operating state (arrow 201 in the chart in Fig. 2a) to the memory controller 3.
35 After this, the processor 4 sends a memory block erasing command to the memory controller 3, and reports the base address, serial number, or corresponding information on the block to be erased, which identifies

the block to be erased in the block memory 2 (arrow 202). In addition, the memory controller 3 is informed preferably of the number of blocks or the length of the memory area to be erased, if there are more than one blocks to be erased. After this, the memory controller 3 sets, if
 5 necessary, the chip enable line 5d to a status, where the connection bus of the block memory is active. This status is, for example, the logic 0-status. When the block memory 2 is activated, the memory controller 3 first examines whether the block memory 2 is ready, for example, to receive commands, or is the block memory 2 in the middle of some
 10 function. If the block memory 2 is performing, for example, programming of data or erasing the memory, the block memory 2 usually cannot receive new commands. This can be determined by examining the status of the ready/busy line 5h. The status can be either ready or busy. Let us assume here that in the ready status the
 15 ready/busy line 5h is in a logic 1-status, and correspondingly in a busy status the ready/busy line is in a logic 0-status.

When the ready/busy line 5h indicates that the block memory 2 is ready to receive commands, the memory controller 3 sets a block
 20 erasing command (arrow 203) to the data bus 5a of the connection bus 5. After this, a pulse is advantageously sent to the command latch enable line 5b, in which case the inner controller 16 of the block memory reads the command from the data bus 5a on the edge of the pulse. Depending on the application, this edge can either be a rising
 25 edge or a downward edge. After this, the memory controller 3 sets an address data of the block to be erased, or a part of it, to the data bus 5a and after the address data has settled on the data bus 5a, the memory controller 3 forms a pulse to the address latch enable line 5c. Thus, the controller 16 of the block memory reads the address data
 30 from the data bus 5a. With the typical 8-bit bus used in this example, it is not possible to address the memory space of the entire block memory 2, in which case the address must be given in several parts. Thus, the memory controller 3 repeats the above described address transmission as many times as necessary, typically two or three times,
 35 in which case the block memory 2 uses all the transmitted address data to identify the block to be erased. When the entire address has been transmitted to the block memory 2, the block memory 2

(advantageously the controller 16) changes the status of the ready/busy line 5h, i.e. sets it to the busy status, for example to a logic 0-status. After this, the block memory 2 begins erasing the memory area, i.e. starts to set all the bits on this memory area to a certain status, advantageously to a logic 1-status. When the ready/busy line 5h is in the busy status, the block memory 2 is performing the set function, i.e. in this case erasing the memory.

10 In connection with erasing the memory area, it is checked in the block memory 2 whether erasing all the bits was successful or not. If some bit has not settled into the desired status, the status data is saved in the block memory 2. After performing the erasing and comparison function, the status of the ready/busy line 5h is reset to the ready status, for example to the logic 1-status (arrow 204). This change in the
15 status of the ready/busy line causes an interruption in the memory controller 3, in which case the memory controller moves to perform a corresponding interruption service program. If an error occurred in erasing, error data of this is transmitted to the memory controller advantageously in the following manner. In the block memory 2, the
20 status of the ready/busy line 5h is changed back to the busy status. Now the memory controller 3 reads the status of the ready/busy line 5h (block 205), and if it is in said busy status, the memory controller 3 can conclude an error has occurred. If necessary, the processor 4 can be informed of this error, for example by an interruption. However, if the
25 status of the ready/busy line is in the ready status, the memory controller 3 can conclude that the function has succeeded and the next command can be transmitted to the block memory 2 (arrow 206). With this arrangement, the memory controller 3 does not have to separately read the status data from the block memory 2, but the function can
30 continue with, for example, erasing the next block, which is depicted by arrows 206 and 207, as well as block 208 in Fig. 2a. In this example situation, the memory controller 3 has detected during the checking 208 of the status data that an error has been detected to have occurred during block erasing in the block memory 2. Thus, the memory
35 controller 3 advantageously forms an interruption (arrow 209) to processor 4, which, on the basis of this interruption, can take the necessary procedures, such as try erasing the block again. If erasing

the block does not succeed despite several erasing attempts, it can be assumed, that in the block in question there is one or more faulty memory cells. Thus, the memory block in question can be marked as unfit for use and no memory procedures are aimed at it after that. It is, however, clear that in some applications the entire block is not necessarily unfit for use, but instead of the entire block it is possible to mark a page where the faulty memory cell is located as unfit for use. In addition, it is possible to arrange error correction function in connection with the block memories, in which case a part of the faulty memory cells can be replaced with another memory cell, or the effect of a faulty memory cell can be attempted to be eliminated with an error correction algorithm or a corresponding method known as such.

Next, storing data to the block memory 2 will be described. This is presented in Fig. 2b as a signalling chart. The processor 4 sends function parameters affecting its operating state (arrow 211 in the chart in Fig. 2b) to the memory controller 3. After this, the processor 4 sends the memory controller 3 the programming command of the memory page, the data to be stored, as well as the address where the data is to be stored in the block memory 2 (arrow 212). This programming address is preferably a page address. As was already stated earlier, an individual byte cannot be stored as such in the block memories, but the data of that entire page where the change is taking place is to be stored. If the data area is larger than one page, the processor 4 informs the memory controller advantageously also, for example, the number of the bytes, pages, or blocks to be stored, or the end address of the area to be stored. On the other hand, if the processor 4 sends all the data to be stored to the memory controller 3, the memory controller can, on the basis of the amount of data it has received, calculate the amount of data to be stored, as well as how many pages there are of the data to be stored.

After this, the memory controller 3 performs the corresponding procedures to the block memory 2, as the ones presented in the erasing example above, i.e. if necessary, sets the status of the chip enable line 5d and examines the status of the ready/busy line 5h. When the ready/busy line 5h indicates that the block memory 2 is

ready to receive commands, the memory controller 3 sets a data programming command (arrow 213) to the data bus 5a of the connection bus 5. After this, advantageously a pulse is sent to the command latch enable line 5b, in which case the inner controller 16 of the block memory 2 reads the command from the data bus 5a on the edge of the pulse. Next, the memory controller 3 sets an address data to the data bus 5a or a part of it, and after the address data has settled on the data bus 5a, the memory controller 3 forms a pulse to the address latch enable line 5c. Thus the controller of the block memory reads the address data from the data bus 5a. After the entire address has been transmitted to the block memory 2, the block memory 2 changes the status of the ready/busy line 5h, i.e. sets it to busy status. After this, the programming of data into the memory cells is started in the block memory 2. When the status of the ready/busy line 5h is in the busy status, the block memory 2 is performing the set function, i.e. in this case saving the data into the memory.

Also, in connection with storing data, it is checked in the block memory 2 whether setting all the bits to the desired status was successful or not. If some bit has not settled into the desired status, the status data is saved in block memory 2. After performing the programming and comparison function, the status of the ready/busy line 5h is reset to the ready status (arrow 214). This change in the status of the ready/busy line causes an interruption at the memory controller 3, in which case the memory controller moves to perform a corresponding interruption service program. If an error occurs in storing, error information of this is transmitted to the memory controller similarly as in connection with the erasing, advantageously in the following manner. In the block memory 2, the status of the ready/busy line 5h is changed back to the busy status. Now the memory controller 3 reads the status of the ready/busy line 5h (block 215) and if it is in said busy status, the memory controller 3 can conclude an error has occurred. However, if the status of the ready/busy line is in the ready status, the memory controller 3 can conclude that the function has succeeded and the next command can be transmitted to the block memory 2. If the data is stored correctly, the data programming can be continued, if necessary. Thus the memory controller 3 sends a programming command, advantageously the

address of the next page, as well as the data to be stored in the page in question (arrow 216). In the method according to an advantageous embodiment of the invention, the memory controller 3 increases the addresses, in which case the processor 4 does not have to take care of it. After the data of a page has been stored in the block memory 2, once again an interruption is formed to the memory controller 3 (arrow 217), which after this checks the status of the ready/busy line (block 218). If an programming error has been detected, the memory controller 3 advantageously forms an interruption (arrow 219) to processor 4, which on the basis of this interruption can take the necessary procedures, such as try storing the data again.

In some memories, data can also be copied inside the memory from one block to some other block without having to transfer the copied data outside the memory. This function is, in some connections, referred to as the Copy-Back technique. In order to perform the copying function advantageously, the processor 4 sends a copy command of the memory page, information on the location of the data to be copied in the memory (for example the base address), as well as the address where the data is meant to be copied in the block memory 2 (for example a side address of the page in question) to the memory controller 3. If the data area to be copied is larger than one page, the processor 4 informs the memory controller advantageously also, for example, the number of the bytes, pages, or blocks to be copied, or the end address of the area to be copied. After receiving the necessary data the memory controller 3 transmits the data to the block memory 3, wherein the copying can be started. In transmitting commands, advantageously the function principles described above in connection with storing are followed. When starting the data copying, the block memory 3 sets the status of the ready/busy line 5h to the busy status. After this, the copying of data is started in the block memory 2. When the ready/busy line 5h is in the busy status, the block memory 2 is performing data copying in the memory. In connection with copying the data, it is checked in the block memory 2 whether copying was successful or not. If an error has occurred in copying, the error data is stored in the block memory 2. After performing the copying and comparison function, the status of the ready/busy line 5h is reset to the

ready status. If an error has occurred in copying, the status of the ready/busy line 5h is changed back into busy status, in which case the memory controller 3 and/or the processor 4 can detect that an error has occurred.

5

The invention can also be applied in connection with a reading data. Even then the data and command/commands required for reading are transmitted to the block memory 2, after which reading the data from the block memory 2 is started. If in connection with reading the block
10 memory 2 detects an error, notification of this is performed, by applying the above-presented principles, by means of status changes. Error detection can be based on, for example, the inner monitoring characteristics possibly in the block memory 2, with which it is possible to examine the working order of the cells.

15

Figure 3 shows, in a reduced manner, the structure of the block memory 2 according to an advantageous embodiment of the invention. The block memory comprises a controller 16, which mostly performs the control of the above-described functions, such as indication of
20 status data, data comparison, controlling the erasing/storing in the block memory 2, etc. In addition, there is an I/O block 17 in the block memory 2 for connecting the block memory 2 to the connection bus 5, a memory cell matrix 18, line and column address buffers 19, 20, a data buffer 21, as well as registers 22.

25

In the electronic device according to Fig. 1, there is also a DMA block 8 (Direct Memory Access). By means of this DMA block 8 it is possible to perform data transmission between the memory 9 of the electronic device and the memory controller 3 without the processor 4 having to
30 take part in the data transmission. Thus, the processor 4 can perform other tasks during data transmission. Thus, the present invention can also be applied in such a manner that the processor 4 starts the data transmission, for example, from the memory 9 to the block memory 2 by applying the principles presented above. The memory controller 3
35 takes care of searching the data from the memory 9 by means of the DMA block 8, of storing the data in the block memory, and of confirming the accuracy of the stored data. In an error situation the

memory controller 3 can, for example, with an interruption inform the processor 4 of an error, in which case the processor 4 can start the procedures caused by the error. On the other hand, the invention can also be applied in such a manner that the memory controller 3 first tries
5 to remove the inaccurately stored data for example by trying the storing again, and if that does not succeed, the memory controller 3 notifies the processor 4 of the error.

Even though the invention is described above in an electronic device,
10 where the memory controller 3 and possibly also the DMA block 8 is used, it is clear that the invention can be applied also without the memory controller 3 and the DMA block 8. If the memory controller 3 is not used, the block memory 2 can be connected to, for example, the bus of the processor 4. Thus, the procedures performed above by the
15 memory controller 3 are performed in the processor 4. Also in this case it is possible to decrease the load of processor 4 with the present invention because, among other things, the status data does not have to be separately read from the block memory 2, but it can be determined by examining the status of the ready/busy line 5h of the
20 block memory.

The above mentioned ready/busy line 5h does not necessarily have to be used as the ready/busy line of the block memory 2, but instead of it some other line of the connection bus 5 can be used, such as some
25 data line of data bus 5a, or it is possible to arrange a specific line for this purpose.

It should be noted here that instead of interruption it is possible to use another method as well, with which the operating state of the block
30 memory can be monitored during data programming and erasing. Thus, for example, the memory controller 3 reads the status of the ready/busy line 5h at intervals and when the memory controller detects a status change from the busy status to the ready status, it is possible to conclude from this that the block memory 2 has finished the function
35 that was started with the previous command. After this, the memory controller 3 can monitor if the block memory 3 changes the status of this ready/busy line relatively quickly back to the busy status, which,

thus, is a sign that an erasing/storing error is detected in block memory 2.

5 After an error situation, the block memory 2 sets the status of the ready/busy line back to the ready status advantageously in that stage when the next operation is directed at the block memory at the latest. The block memory can detect this from the change of status of some control line, such as the status change in the chip select line 5d.

10 In the electronic device 1 according to Fig. 1 there is also a user interface, wherein there is e.g. a display 10, a keyboard 11, an earpiece 12, and a microphone 13. In addition, in the electronic device 1 there is also a transmitter 14 and a receiver 15 for performing mobile communication functions.

15 The invention can also be applied in connection with memory cards (not shown). This type of memory cards can be connected by means of a card interface arranged to the electronic device 1. Thus, the transmission of commands and signals as presented above is
20 performed via the card interface. Block memory applying the method according to the invention is advantageously arranged to the memory card.

25 It will be obvious that the present invention is not limited solely to the above-presented embodiments but it can be modified within the scope of the appended claims.

Claims:

1. A method for performing a processing function of a memory in a block memory (2), wherein there are memory cells (18), where data
5 can be stored, and a connection bus (5), wherein there is at least a ready/busy line (5h), which can be set at least to a ready status and to a busy status, and in which method setting the status of said ready/busy line (5h) to said busy status is performed in the beginning of the processing function, and detection of processing errors is
10 performed, and the end of processing function is indicated by setting said ready/busy line (5h) to said ready status, **characterized** in that if a processing error is detected, the status of said ready/busy line is changed back to said busy status.
- 15 2. The method according to claim 1, **characterized** in that said processing function is erasing the block memory (2), in which case a comparison of the erased memory cell with the data meant for the status of the erased memory cell is performed in connection with erasing, in order to detect erasing errors, and if an erasing error is
20 detected in the comparison, the status of said ready/busy line is changed back to said busy status.
- 25 3. The method according to claim 1, **characterized** in that said processing function is storing data to the block memory (2), in which case the comparison of the stored data with the data meant to be stored is performed in connection with storing the data, in order to detect storing errors, and if a storing error is detected in the comparison, the status of said ready/busy line is changed back to said
30 busy status.
- 35 4. The method according to claim 1, **characterized** in that said processing function is copying data in the block memory (2) from one location to some other location, in which case the comparison of the copied data with the data meant to be copied is performed in connection with copying the data, in order to detect copying errors, and if a copying error is detected in the comparison, the status of said ready/busy line is changed back to said busy status.

5. The method according to claim 1, **characterized** in that said processing function is reading data from the block memory (2), in which case examining the status of the memory cell being read is performed
5 in connection with reading the data, and if a memory cell error is detected in the examination, the status of said ready/busy line is changed back to said busy status.
6. The method according to any of the claims 1 to 5, **characterized** in
10 that the processing function is started by sending a command to the block memory (2), and that the status of said ready/busy line (5h) is set to the busy status in the stage when the function according to the command is started in the block memory (2).
7. The method according to any of the claims 1 to 6, **characterized** in
15 that the block memory (2) is divided into blocks, and each block is divided into pages, in which case data is transmitted to the block memory (2) page by page.
8. The method according to claim 7, **characterized** in that erasing of at
20 least one block is performed, in which case a previously determined status is set as the status of all the memory cells of said at least one block.
9. The method according to claims 1 to 8, **characterized** in that the
25 status of said ready/busy line (5h) is examined, in which case when it changes from the busy status to the ready status, re-examination of the status of the ready/busy line (5h) is performed.
10. The method according to any of the claims 1 to 9, **characterized**
30 in that in the connection bus (5) there is a data bus (5a) comprising at least one data line, and that the data line of said data bus (5a) is used as a ready/busy line (5h).
11. A system, which comprises an electronic device (1), wherein there
35 is block memory (2), wherein there are memory cells (18), where data is arranged to be stored, and a connection bus (5), wherein there is at

least a ready/busy line (5h), which is arranged to be set at least to a ready status and to a busy status, and which block memory (2) comprises means (16) for changing the status of said ready/busy line (5h) to said busy status in the beginning of the processing function of the block memory, a comparator (16) for detecting processing errors, and the end of processing function is arranged to be indicated by setting said ready/busy line (5h) to said ready status, **characterized** in that after detecting a processing error the status of said ready/busy line (5h) is arranged to be changed back to said busy status.

12. The system according to claim 11, **characterized** in that said processing function is one of the following:

- emptying the memory cells (18),
- storing data in memory cells (18),
- copying data between memory cells (18),
- reading the data stored in the memory cells (18).

13. The system according to claim 11 or 12, **characterized** in that it comprises a processor (4), and a first connection bus between the processor (4) and the block memory (2), in which case the processing function is arranged to be started by sending a command to the block memory (2), and that the status of said ready/busy line (5h) is arranged to be set to the busy status when the function according to the command has been started in the block memory (2).

14. The system according to claim 13, **characterized** in that it comprises means (3, 5, 7) for creating an interruption in the processor (4) when the status of the ready/busy line (5h) changes from the busy status to the ready status, in which case the status of the ready/busy line (5h) in connection with handling the interruption is arranged to be examined in the processor (4).

15. The system according to claim 11 or 12, **characterized** in that it comprises a processor (4), a memory controller (3), a first connection bus between the processor (4) and the memory controller (3), and a second connection bus between the memory controller (3) and the block memory (2), in which case the commands are arranged to be

sent from the processor (4) to the memory controller, as well as from the memory controller to the block memory (2), in which case storing the data is arranged to be started by sending a command to the block memory (2), and that the status of said ready/busy line (5h) is arranged to be set to the busy status when the function according to the command has been started in the block memory (2).

16. The system according to claim 15, **characterized** in that it comprises means (5) for creating an interruption in the memory controller (3) when the status of the ready/busy line (5h) changes from the busy status to the ready status, in which case the status of the ready/busy line (5h) in connection with handling the interruption is arranged to be examined in the memory controller, and if the ready/busy line is in the busy status an interruption to the processor (4) is arranged to be formed from the memory controller (3).

17. The system according to any of the claims 11 to 16, **characterized** in that the block memory (2) is divided into blocks, and each block is divided into pages, in which case data is arranged to be transmitted to the block memory (2) page by page.

18. The system according to any of the claims 11 to 17, **characterized** in that in the bus interface (5) there is a data bus (5a) comprising at least one data line, and that the data line of said data bus (5a) is arranged to be used as a ready/busy line (5h).

19. An electronic device (1), which comprises block memory (2), wherein there are memory cells (18), where data is arranged to be stored, and a bus interface (5), wherein there is at least a ready/busy line (5h), which is arranged to be set at least to a ready status and to a busy status, and which block memory (2) comprises means (16) for setting the status of said ready/busy line (5h) to said busy status in the beginning of the processing function of the block memory, a comparator for detecting processing errors, and the end of processing function is arranged to be indicated by setting said ready/busy line (5h) to said ready status, **characterized** in that after detecting a processing

error, the status of said ready/busy line (5h) is arranged to be changed back to said busy status.

5 20. A block memory (2), wherein there are memory cells (18), where data is arranged to be stored, and a connection bus (5), wherein there is at least a ready/busy line (5h), which is arranged to be set at least to a ready status and to a busy status, and which block memory (2) comprises means (16) for setting the status of said ready/busy line (5h) to said busy status in the beginning of the processing function of the
10 block memory, a comparator for detecting processing errors, and the end of processing function is arranged to be indicated by setting said ready/busy line (5h) to said ready status, **characterized** in that after detecting a processing error, the status of said ready/busy line (5h) is arranged to be changed back to said busy status.

15 21. The memory block (2) according to claim 20, **characterized** in that said processing function is one of the following:

- emptying the memory cells (18)
- storing data in memory cells (18),
- 20 - copying data between memory cells (18),
- reading the data stored in the memory cells (18).

22. The block memory (2) according to claim 20 or 21, **characterized**
25 in that it comprises a comparator (16) for comparing the data stored in the memory cells (18) with the data meant to be stored in order to detect storing errors.

(57) Abstract:

The invention relates to a method for performing a memory processing function to a block memory (2), wherein there are memory cells (18), where data can be stored, and a connection bus (5), wherein there is at least a ready/busy line (5h). The ready/busy line (5h) can be set at least to a ready status and a busy status. In the method, setting the status of said ready/busy line (5h) to said busy status is performed at the beginning of the processing function, and the detection of processing errors is performed. The end of processing function is indicated by setting said ready/busy line (5h) to said ready status. After detecting a processing error, the status of said ready/busy line (5h) is changed back to said busy status if an processing error is detected. The invention also relates to a system, an electronic device (1) and a block memory (2), in which the method is applied.

Fig. 2a

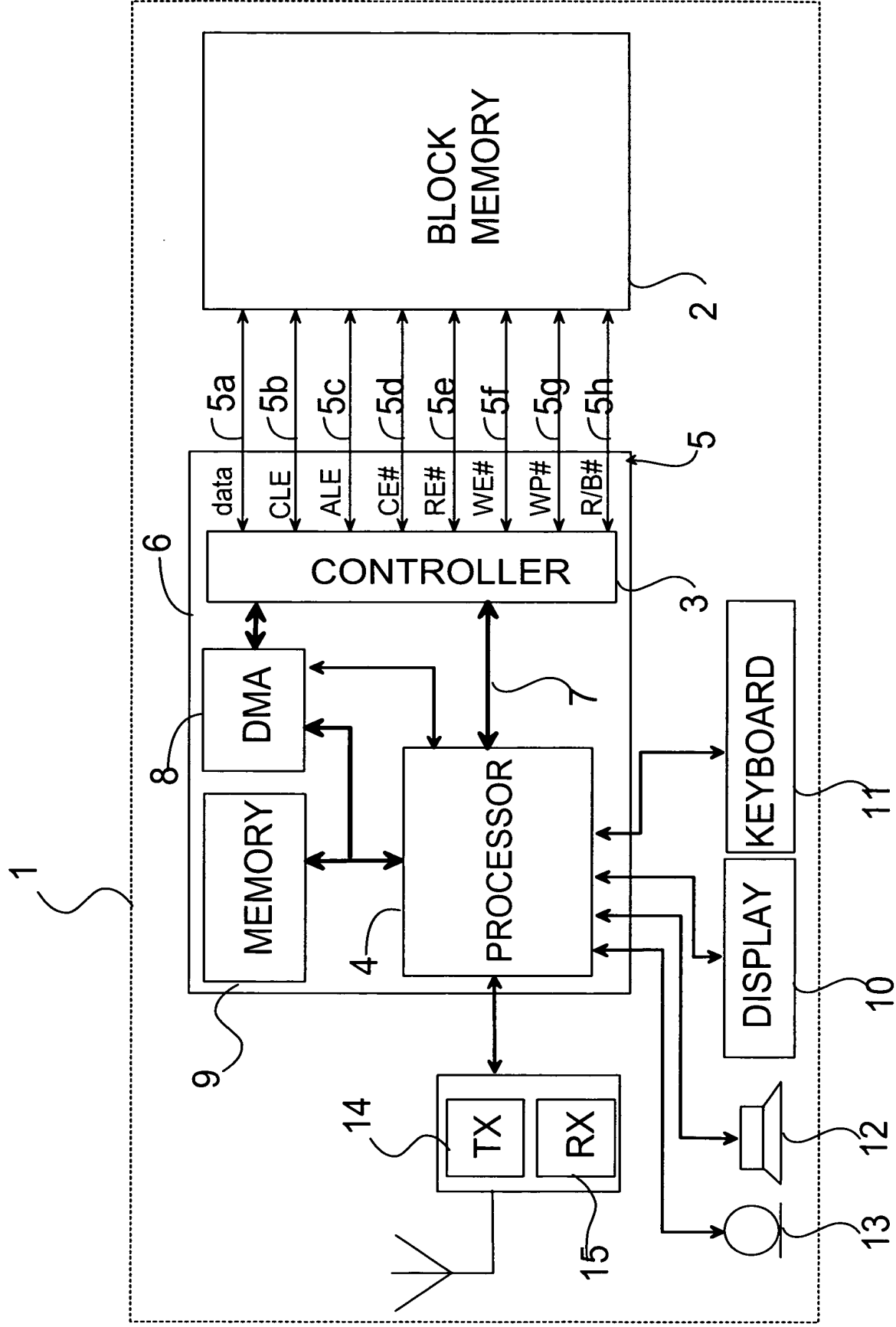


Fig. 1

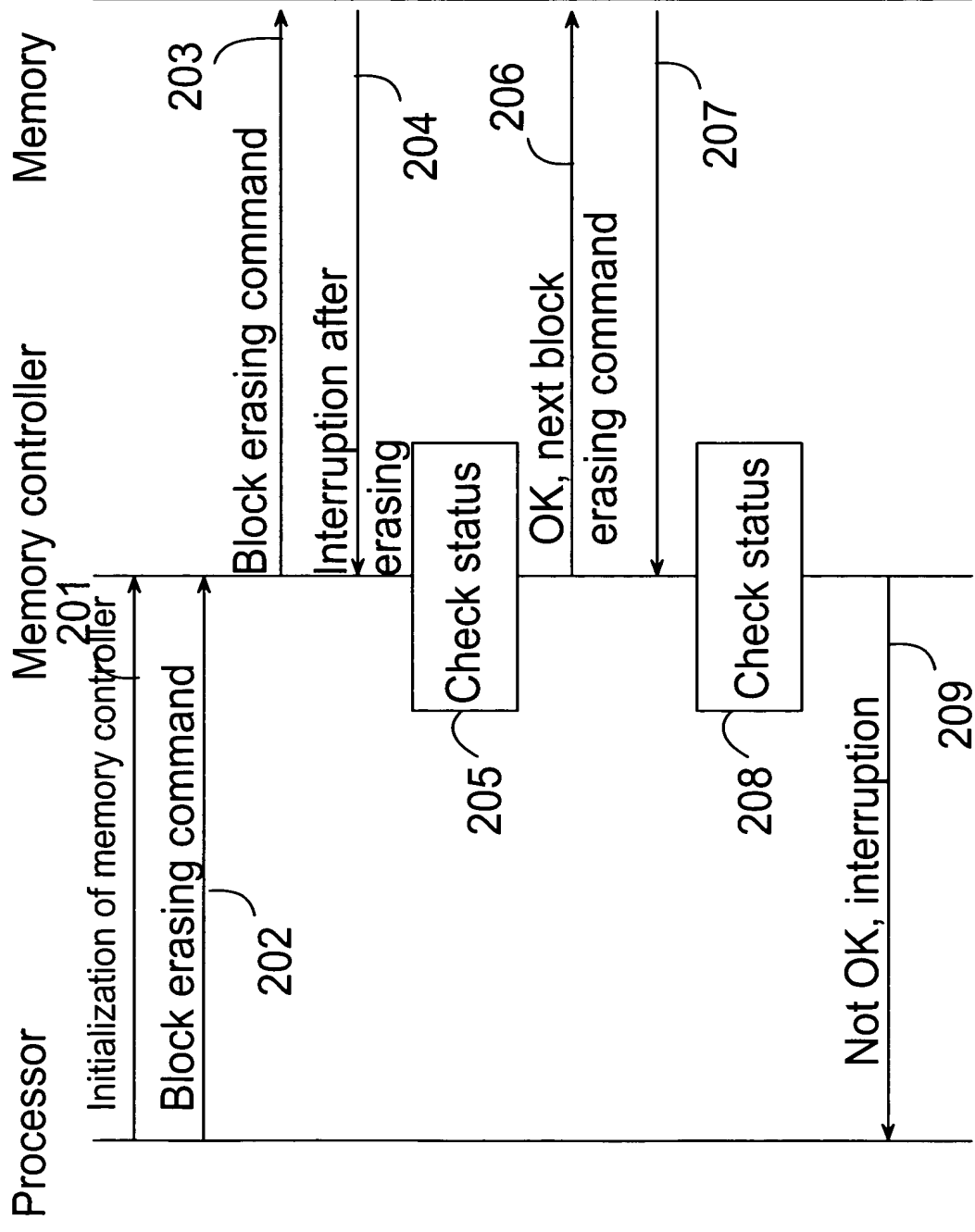


Fig. 2a

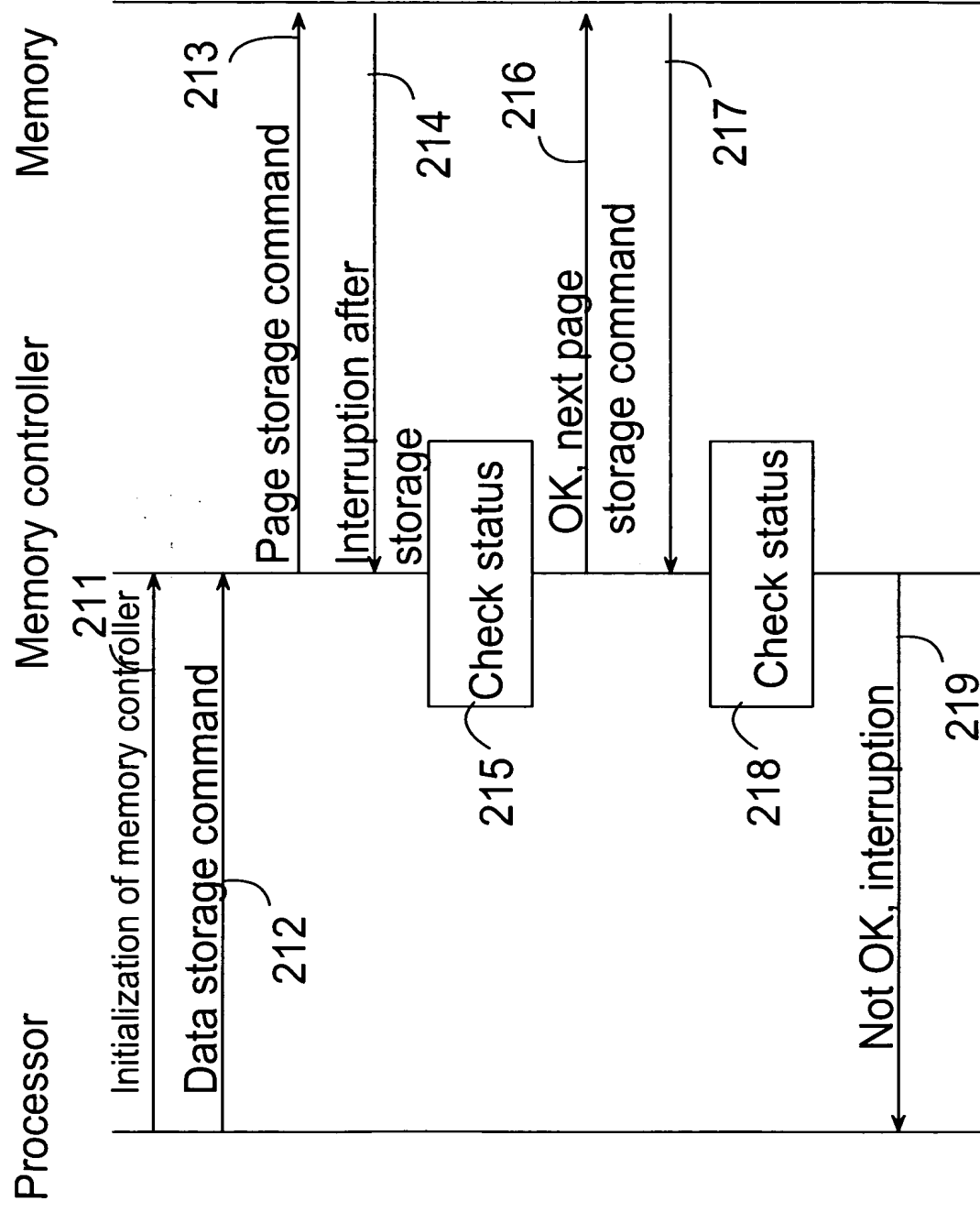


Fig. 2b

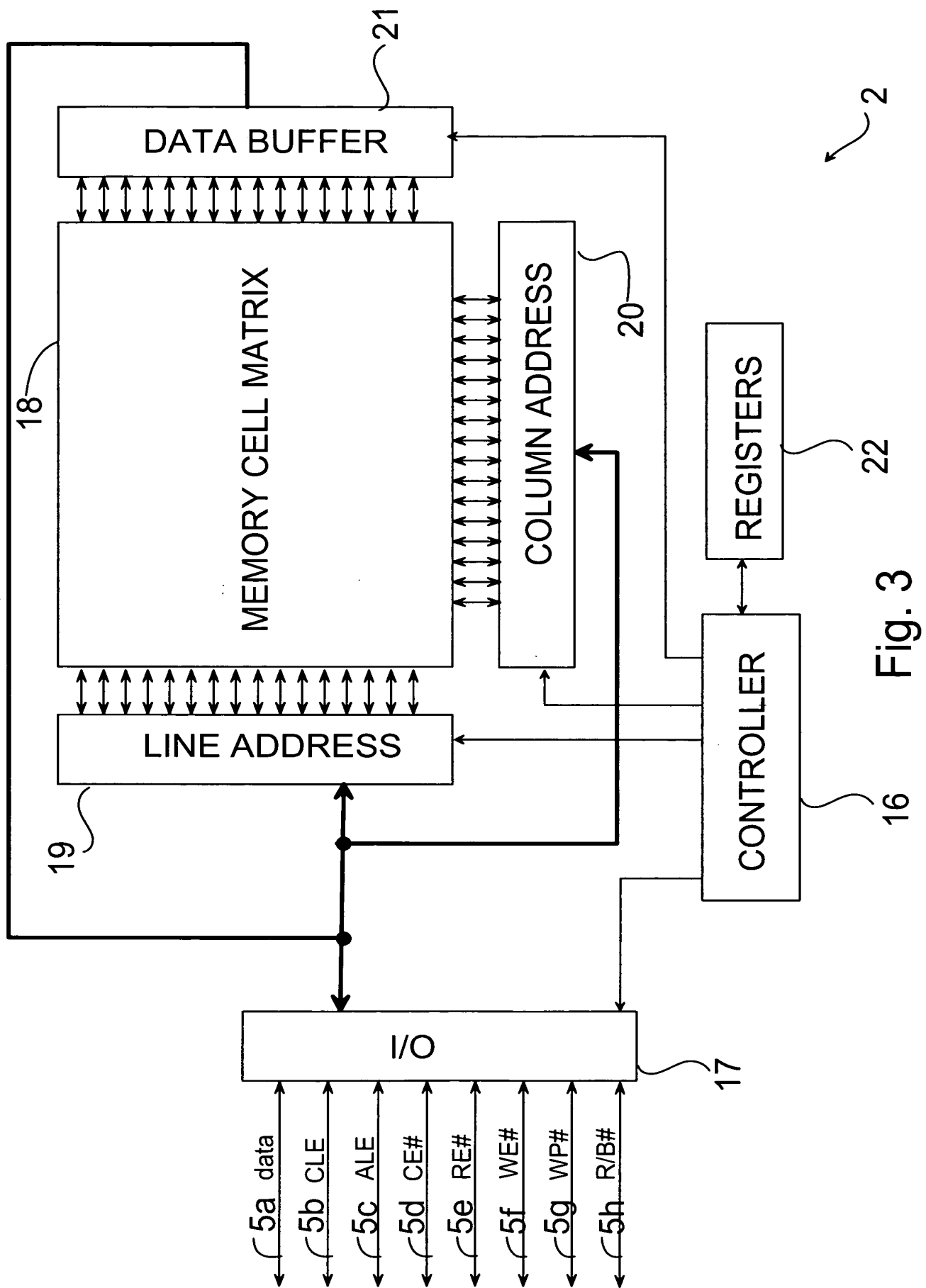


Fig. 3